Residential Systems LEAP Protocol - Control and Monitoring

Document Change History

LEAP Version	Author	Date	Comments	Released
1.120	Christie Evans / John Nill	06 Aug 2019	• Initial Release	❌
3.0	Anuj Chheda	06 Feb 2020	• Initial Release for HQWSX	❌
3.0	Deeksha Yogish / Dazzy Gupta (Deactivated) / Ryan O'Donnell	30 Apr 2020	• Addressed comments on the initial release document	❌
3.1	Deeksha Yogish	13 Jul 2020	• Added AreaScenes • Added Permissions	❌
3.2	Former user (Deleted)	25 Aug 2020	• Added Area Status	❌
3.201	Deeksha Yogish	01 Oct 2020	• Added ○ GoToGroupLighting ○ PresetActivate	
3.203	Dazzy Gupta (Deactivated)	12 Jan 2021	• Added ○ GoToScene ○ WhiteTuningLevelAssignment preset filter read ○ GoToWhiteTuningLevel • Removed ○ PresetActivates for Username/Password integrators	

3.204	Former user (Deleted)	01 Mar 2021	• Added ○ TimeclockEvent ○ SystemLoadSheddingStatus	
3.205	Dazzy Gupta (Deactivated)	23 Mar 2021	• Added OpenCloseStop in ControlType enum	
3.205	Dazzy Gupta (Deactivated)	07 Apr 2021	• Removed ProgrammingModel bulk read for Homeworks	
3.205	Deeksha Yogish	16 Aug 2021	• Added HVACs	
3.212	Anuj Chheda	10 Nov 2021	• Added RadioRA 3	
3.216	Charlie Kelley	04 Feb 2022	• Added ○ versioned read on spectrumtuninglevelassignment to include naturalshow	
3.225	Anuj Chheda	14 Jun 2022	• Added ButtonGroupExpanded	
3.231	Jake Recoulle/ Deeksha Yogish	08 May 2023	• Added Subscribe support for ○ Areas ○ Control Stations ○ Zones • Added ColorTuningProperties to the Zone resource	
3.234	Jake Recoulle	18 Jul 2023	• Added ○ CurveDimming Definition ○ WarmDimAssignments preset Filtered Read ○ GoToWarmDim Command • Modified Zone Resource ○ Added AvailableControlTypes ○ Added DefaultDimCurve	

3.234	Jake Recoulle	24 Jul 2023	• Added example Zone Read on a WhiteTune and Warm Dim capable zone • Marked ControlType as deprecated, left note in description • Add Fade and Delay to GoToWarmDim example	
3.242	Jesse Rittner / Marc Ferdinandusse	06 Aug 2024	• Split AvailableControlType enum from ControlType enum • GoToDimmedLevel command changes for all dimmable zones	

Table of Contents

Usage of the LEAP integration protocol assumes that the developer creating the integration script understands how to establish TLS connections, how to create and parse JSON messages and how to process standard HTTP response codes.

The Lutron Extensible Application Protocol (LEAP) will allow third-party equipment, such as touch-screens, universal remote controls, and software applications, to control and monitor devices in a Lutron system. The protocol is supported by Caséta , RA2 Select, HomeWorks and RadioRA 3 systems. In each of these systems, there is a bridge, processor or Main Repeater, which acts as the central controller. In the LEAP protocol, this is referred to as the "Master Device".

Using LEAP to Integrate

Connections and Authentication

There are two distinct ways of integrating with Lutron

1. Cert-based integrations
2. Username/Password integrations

Lexicon

Term	Definition
Server	Lutron Product
Client	Integration
LAP	Local Association Protocol connection Pre-cursor to establishing a LEAP connection. All cert-based integrations are required to authenticate themselves to the Lutron product via the Local Association Protocol before attempting to be able to establish a LEAP connection.

Ethernet Bridge Discovery

The Ethernet Bridge runs a zeroconf service to allow other computers to discover it on the network. The Ethernet Bridge is running an Avahi service that is discoverable using standard zerconf clients such as Apple Bonjour, mzclient, etc. The Ethernet Bridge will publish one DNS-SD service. The service name is “lutron” and the protocol is “tcp” so the name will be seen as “_lutron_tcp”. The accompanying TXT record will have a DEVCLASS key that can be used to determine if the type of the Lutron Bridge. Use the table below if you need to filter for specific types of products as there may be multiple on one network.

Product	Device Class
Smart Bridge	Starts with 0x0804
Smart Bridge Pro	Starts with 0x0805
Connect Bridge	Starts with 0x0809
RA2 Select Main Repeater	Starts with 0x080E

Connections and Authentication

The LEAP server can support 10 concurrent client connections, shared across local and remote. For example, if 8 clients are connected locally, and 2 are connected remotely, another attempt to connect remotely would result in a MaxConnections error, and the connection being closed. The same would happen if an additional local connection were attempted to be made.

The MaxConnections error is:

MaxConnections Error

```
{
    "CommuniqueType": "ExceptionResponse",
    "Header": {
        "MessageBodyType": "ExceptionDetail",
        "StatusCode": "503 ServiceUnavailable"
    },
    "Body": {
        "Message": "The max number of clients are connected. You will need to
wait until someone disconnects before attempting to connect again."
    }
}
```

General Notes when Sending LEAP Messages

- The first message that is sent after a successful connection **must** always be to negotiate the major protocol version. This will avoid any unnecessary versioning errors.
- All requests and responses must be terminated with a carriage return, line feed (\r\n).
- All responses are either resource definitions or hyper-references. When a resource definition contains references to other resources, hyper-references are returned. A hyper-reference contains only the URL for the resource. A definition contains all properties of the resource.
- Unexpected properties must be ignored by the client.
- All client requests will generate a response from the server. If there is an issue with the request (eg the request is malformed, or not supported) an ExceptionResponse will be returned.
- Only **one** message will be processed by the server at a time. For eg, if a client commands a switched zone to a level using GoToSwitchedLevel, and then issues a ping request, the ping response will not be returned until the response for the GoToSwitchedLevel_command is returned.
- Some requests, such as SubscriptionRequests, may generate asynchronous responses. Other requests which may take longer to complete may return a 102 Processing immediately, followed by an asynchronous completion response. For these messages, it can be beneficial to use the ClientTag property in the header to pair a request with its response. ClientTag is an optional string parameter in a request which will be echoed in any response generated by that request. For instance, a ClientTag specified in a SubscriptionRequest will be echoed in all notifications generated by that subscription.
- Unless otherwise noted, the order of elements in a list is arbitrary and could change between responses

LEAP Lexicon

Request

- Requests are always sent from the client to the server. A request may or may not require a body depending on the `CommuniqueType`.
- The client is required to delimit all requests with a newline (\n) or a carriage return followed by a newline (\r\n). These characters are not permitted to appear within a request.
- The client should not send unnecessary whitespace characters within a request, but the server must ignore them if present.
- Empty properties (i.e. empty lists, or empty strings) might be omitted by the server.

Response

- Responses are always sent from the server to the client. Responses may either be synchronous (i.e., sent as the direct response to a request) or asynchronous (i.e., sent in response to some system event).
- A response may or may not require a body depending on the `StatusCode` and `MessageBodyType`.
- The server is required to delimit all responses with a carriage return followed by a newline (\r\n). These characters are not permitted to appear within a response.
- The server should not send unnecessary whitespace characters within a response, but the client must ignore them if present.

Backwards Compatibility

Changes to the major version of LEAP reflect breaking changes in the protocol. In order to maintain backwards compatibility, all remote clients MUST update "/clientsetting" to the desired major version of LEAP in their first request and interpret the response appropriately. See the ClientSetting documentation for more information.

Changes to the minor version of LEAP reflect non-breaking changes in the protocol, such as the addition of new features. Clients need not worry if the server reports a minor version greater than what they need. Note that the following are NOT considered breaking changes:

- adding a new value to an existing enum
- adding a new `ErrorCode` to an existing `ExceptionResponse`
- adding a new interim 1xx response for an existing request
- using a new or different 2xx, 4xx, or 5xx `StatusCode` for an existing response
- using a non-204 2xx `StatusCode` for an existing response that previously only used `"204 NoContent"`
- adding new fields to an existing resource
- adding new asynchronous notifications (e.g., a status notification containing only a new field that the client doesn't recognize)

URLs

Most URLs in LEAP are an implementation detail of the server. The client must not attempt to generate or validate any URLs on their own. As an exception to this, there are a handful of guaranteed URLs that the client can generate. These are recorded throughout this documentation.

Note that within the request or response header, the unique identifier of a resource is called the `Url`. However, everwhere else it is called the `href`.

ClientTag

The client may optionally include a `ClientTag` in the header of any request. If they do, it will be echoed in the header of the corresponding response, even if it is an `ExceptionResponse`.

Additionally, if a `ClientTag` is provided with a `SubscribeRequest`, it will be echoed in every asynchronous notification for that subscription.

StatusCode

The `Header` of a response always contains a `StatusCode` indicating the success or failure of a request. These status codes are all based on HTTP status codes. A 2xx code indicates success, while a 4xx or 5xx code indicates failure.

A 1xx code indicates an interim response that implies neither the success nor failure of the request. The server may send any number of such interim responses before the final response, and the client must ignore all interim responses that they are not expecting. The server must still send a final response with a non-1xx status code.

MessageBodyType

If a response contains a `Body`, its `MessageBodyType` will contain a string denoting its structure. This string typically is comprised of three pieces. The first is the plurality - `One` or `Multiple`. The second is the resource type, such as `Area` or `Device`. The third is the resource category, which is almost always `Definition` or `Status`.

For simplicity, whenever this documentation says that the response `Body` will be a `OneFooDefinition`, for example, that implies that the `MessageBodyType` will also be the string `"OneFooDefinition"`.

Omitting Properties

The server reserves the right to omit properties that are empty. For example, an empty list or object may be omitted entirely from the response.

When reading a list resource, if the list is empty, the entire `Body` may be omitted. When this occurs, the `MessageBodyType` will be omitted, and the `StatusCode` will be set to `"204 NoContent"`. For example, if a client asked to read the list of all areas, and there were no areas in the system, the server could respond with `{"CommuniqueType":"ReadResponse","Header":{"Url":"/area","StatusCode":"204 NoContent"}}`.

Errors

If the server encounters an error processing a request, it will respond with an `ExceptionResponse`. See ExceptionResponses for more information.

Pings

The server will close any connections that have been inactive for at least 3 minutes. Clients should ping the server every 30 seconds if no other requests are being sent. Note that only receiving requests qualifies as activity, not sending responses, including asynchronous notifications. See Keeping Connections Alive for more information.

Status Resource

At a protocol level, all definition resources have a corresponding status resource. If a particular status resource is supported, its href is guaranteed to be the href of the definition resource, with "/status" appended to it. For example, if the href of an area were "/area/a", then the href of its status would be "/area/a/status".

Command Processor

At a protocol level, all resources have a command processor. If a particular command processor is supported, its href is guaranteed to be the href of the resource, with "/commandprocessor" appended to it. For example, if the href of an area were "/area/a", then the href of its command processor would be "/area/a/commandprocessor".

When sending a request to a command processor, send a `CreateRequest` with a `Body` of type `OneCommandDefinition`. Unless otherwise noted, the response is always a `CreateResponse` with a `StatusCode` of `201 Created`.

Concurrency

The LEAP server is single-threaded per client connection. Consequently, clients should avoid sending a second request until the response to the first request has been received. Note that this does not apply to asynchronous notifications. The server is free to send such responses at any time, including between a request and its synchronous response.

Names

Many resources have a `Name` property. Unless otherwise noted, all names must not exceed 50 bytes in length (using UTF-8 encoding) and must not contain double quotes ("). Note that this restriction applies only to the Rockhopper LEAP server, not the general LEAP protocol.

Operations

LEAP supports the standard create, read, update, and delete operations. In addition, it supports subscribing to resources to receive asynchronous notifications when their representation changes.

Create

To create a new resource, send a request with `CommuniqueType` set to `CreateRequest`. Typically, the request must be issued with `Header.Url` set to the applicable list URL. A `CreateRequest` always requires a valid `Body`. Unless otherwise noted, the response to a `CreateRequest` is always a `CreateResponse` with a `StatusCode` of `201 Created` and a `Body` containing the representation of the new resource.

Clients should not include any properties in a `CreateRequest` that will be ignored by the server. Otherwise, they might break if a future server implementation does consider that property. This includes properties that are not currently part of the specification.

Example

Request

```
{
    "CommuniqueType":"CreateRequest",
    "Header":{
        "Url":"/foo"
    },
    "Body":{
        "Foo":{
            "Name":"My Name"
        }
    }
}
```

Response

```
{
    "CommuniqueType":"CreateResponse",
    "Header":{
        "MessageBodyType":"OneFooDefinition",
        "StatusCode":"201 Created",
        "Url":"/foo"
    },
    "Body":{
        "Foo":{
            "href":"/foo/1",
            "Name":"My Name"
        }
    }
}
```

Read

To read a resource, send a request with `CommuniqueType` set to `ReadRequest` and `Header.Url` set to the URL of the resource. A `ReadRequest` always requires that `Body` be omitted. Unless otherwise noted, the response to a `ReadRequest` is always a `ReadResponse` with a `StatusCode` of `200 OK` and a `Body` containing the representation of the resource.

Example

Request

```
{
    "CommuniqueType":"ReadRequest",
    "Header":{
        "Url":"/foo/1"
    }
}
```

Response

```
{
    "CommuniqueType":"ReadResponse",
    "Header":{
        "MessageBodyType":"OneFooDefinition",
        "StatusCode":"200 OK",
        "Url":"/foo/1"
    },
    "Body":{
        "Foo":{
            "href":"/foo/1",
            "Name":"My Name"
        }
    }
}
```

Update

To update a resource, send a request with `CommuniqueType` set to `UpdateRequest` and `Header.Url` set to the URL of the resource. An `UpdateRequest` always requires a valid `Body`. Unless otherwise noted, the response to an `UpdateRequest` is always an `UpdateResponse` with a `StatusCode` of `200 OK` and a `Body` containing the new representation of the resource.

Note that all updates in LEAP are partial updates, which means that the client only needs to send those properties they wish to change in the request. Any property omitted from the request will not be changed.

Clients should not include any properties in an `UpdateRequest` that will be ignored by the server. Otherwise, they might break if a future server implementation does consider that property. This includes properties that are not currently part of the specification.

Example

Request

```
{
    "CommuniqueType":"UpdateRequest",
    "Header":{
        "Url":"/foo/1"
    },
    "Body":{
        "Foo":{
            "Name":"My New Name"
        }
    }
}
```

Response

```
{
    "CommuniqueType":"UpdateResponse",
    "Header":{
        "MessageBodyType":"OneFooDefinition",
        "StatusCode":"200 OK",
        "Url":"/foo/1"
    },
    "Body":{
        "Foo":{
            "href":"/foo/1",
            "Name":"My New Name"
        }
    }
}
```

Delete

To delete a resource, send a request with `CommuniqueType` set to `DeleteRequest` and `Header.Url` set to the URL of the resource. A `DeleteRequest` always requires that `Body` be omitted. Unless otherwise noted, the response to a `DeleteRequest` is always a `DeleteResponse` with a `StatusCode` of `204 NoContent`.

Example

Request

```
{
    "CommuniqueType":"DeleteRequest",
    "Header":{
        "Url":"/foo/1"
    }
}
```

Response

```
{
    "CommuniqueType":"DeleteResponse",
    "Header":{
        "StatusCode":"204 NoContent",
        "Url":"/foo/1"
    }
}
```

Subscribe

To subscribe to a resource, send a request with `CommuniqueType` set to `SubscribeRequest` and `Header.Url` set to the URL of the resource. A `SubscribeRequest` always requires that `Body` be omitted. Unless otherwise noted, the response to a `SubscribeRequest` is always identical to a `ReadResponse`, except the `CommuniqueType` will be `SubscribeResponse`.

Multiple subscriptions to the same resource from the same client are not supported. If a client subscribes to the same resource twice, the original subscripion will be discarded, and the new one will take its place.

Unless otherwise noted, whenever the resource to which the client has subscribed changes, they will receive an asynchronous `ReadResponse` with the new representation. Note that the exact behavior of a subscription varies depending on whether the resource in question is a definition, a status or an event.

If the resource to which the client subscribed is deleted, they will receive an asynchronous `DeleteResponse`. Note that no asynchronous notification is sent when a client manually unsubscribes.

Definitions

The entire definition will be transmitted whenever anything in the definition changes. The client is responsible for calculating deltas if required.

Statuses

The server may only include properties that have changed. Note that it will still include static, identifying information, such as the resource's `href`.

For lists, the client may receive a list only containing those resources whose statuses have changed. There is no provision for notifying the client when one of those resources gets deleted. Instead, clients should subscribe to the corresponding definition list if they need to know such information.

Events

The server may only include properties pertaining to the specific event. Note that it will still include static, identifying information, such as the resource's `href`.

For lists, the client may receive a list only containing those resources whose statuses have changed. There is no provision for notifying the client when one of those resources gets deleted. Instead, clients should subscribe to the corresponding definition list if they need to know such information.

Example

Request

```
{
    "CommuniqueType":"SubscribeRequest",
    "Header":{
        "Url":"/foo/1"
    }
}
```

Response

```
{
    "CommuniqueType":"SubscribeResponse",
    "Header":{
        "MessageBodyType":"OneFooDefinition",
        "StatusCode":"200 OK",
        "Url":"/foo/1"
    },
    "Body":{
        "Foo":{
            "href":"/foo/1",
            "Name":"My Name"
        }
    }
}
```

Notification of Change

```
{
    "CommuniqueType":"ReadResponse",
    "Header":{
        "MessageBodyType":"OneFooDefinition",
        "StatusCode":"200 OK",
        "Url":"/foo/1"
    },
    "Body":{
        "Foo":{
            "href":"/foo/1",
            "Name":"My New Name"
        }
    }
}
```

Notification of Deletion

```
{
    "CommuniqueType":"DeleteResponse",
    "Header":{
        "StatusCode":"204 NoContent",
        "Url":"/foo/1"
    }
}
```

Unsubscribe

To unsubscribe from a resource, send a request with `CommuniqueType` set to `UnsubscribeRequest` and `Header.Url` set to the URL of the resource. An `UnsubscribeRequest` always requires that `Body` be omitted. Unless otherwise noted, the response to an `UnsubscribeRequest` is always an `UnsubscribeResponse` with a `StatusCode` of `204 NoContent`.

Example

Request

```
{
    "CommuniqueType":"UnsubscribeRequest",
    "Header":{
        "Url":"/foo/1"
    }
}
```

Response

```
{
    "CommuniqueType":"UnsubscribeResponse",
    "Header":{
        "StatusCode":"204 NoContent",
        "Url":"/foo/1"
    }
}
```

LEAP Resources

In LEAP the various parts of the system are represented as "Resources". These represent system objects that may be controlled or monitored.

Negotiating Protocol Version(ClientSetting)

Integrator Type	Supported
Cert-Based	✓
Username/Password	✓

When a new LEAP major version is released, the older revisions will be supported for some time. Upon connecting to a system, the client's first request must be for the major version the client wishes to use. This is done by sending an Update Request to the server with the LEAP version that you would like to use. Setting the LEAP version ensures the commands and responses will continue to be compatible with your script as the version of Lutron system advances. The table below calls out what versions are supported by each system.

System	Version supported
Caséta	1.x
RA2 Select	1.x
HomeWorks	3.x
RadioRA 3	3.x

If the LEAP server responds with an ExceptionResponse with error code 2, this means that the requested LEAP major version is too low. LEAP clients that request a major version above what is supported by the server will receive an UpdateResponse with the highest supported major version.

Property	Required	Update-able	Type	Details
href	✓		string	The unique identifier for the resource.
ClientMajorVersion	✓	✓	int	The major version of the LEAP protocol which the client wishes to use for the remainder of this connection. Refer table above for supported versions.

ClientMinorVersion			int	The property cannot be updated. If provided in an update request, an exception will be returned.
Permissions	✓		Permissions	Contains information about the Permissions that are assigned to the specific client

Property	Required	Update-able	Type	Details
SessionRole	✓		enum	SessionRole applies to the current session

Session Role enum

Value	Details
Admin	Access to all LEAP resources (for cert-based integrators)
ControlAndEdit	After successful login, allows control, editing and monitoring of major non-admin features
ControlAndMonitor	After successful login, allows control, and monitoring of major non-admin features
Monitor	After successful login, allows only monitoring of major non-admin features
Unauthorized	Not authorised

Client sends Update Request:

```
{
    "CommuniqueType": "UpdateRequest",
    "Header": {
        "Url": "/clientsetting"
    },
    "Body": {
        "ClientSetting": {
            "ClientMajorVersion": 1
        }
    }
}
```

Server Response:

```
{
  "CommuniqueType": "UpdateResponse",
  "Header": {
    "MessageBodyType": "OneClientSettingDefinition",
    "StatusCode": "200 OK",
    "Url": "/clientsetting"
  },
  "Body": {
    "ClientSetting": {
      "href": "/clientsetting",
      "ClientMajorVersion": 3,
      "ClientMinorVersion": 0,
      "Permissions": {
        "SessionRole": "Admin"
      }
    }
  }
}
```

Keeping Connections Alive(Ping Requests)

Integrator Type	Supported
Cert-Based	✓
Username/Password	✓

The server will assume a disconnected client if it does not receive any requests for 3 minutes. If no actual requests need to be sent, a Ping Request can be issued every 30 seconds instead to ensure the connection is maintained. If the anticipated server response is not received, you should re-establish connection with the Lutron system.

Client Request

```
{
    "CommuniqueType": "ReadRequest",
    "Header": {
        "Url": "/server/status/ping"
    }
}
```

Server Response:

```
{
    "CommuniqueType": "ReadResponse",
    "Header": {
        "MessageBodyType": "OnePingResponse",
        "StatusCode": "200 OK",
        "Url": "/server/status/ping"
    },
    "Body": {
        "PingResponse": {
                              "LEAPVersion":3
               }
    }
}
```

Exception Responses

The CommuniqueType will always be "ExceptionResponse" regardless of the action (Read, Create, etc) that created it. The Header will contain the Url of the request, the ClientTag from the request, and will have an ExceptionDetail MessageBodyType.

The status code of the exception response should be consulted first to understand the type of exception. Standard HTTP status codes are used, including:

- **Status codes 400 to 499 - User Error**

Error	Reason
400 BadRequest	There was an issue parsing the request (e.g. the request was not valid JSON, the request is not supported, etc.). Use the Message property to get more details.
401 Unauthorized	An unauthorized request is received. For eg, if a username/password integrator sends LEAP requests before logging in with the correct set of credentials.
404 NotFound	The URL in the request is not recognized by the system. For example, if a read request is issued on /device/5 and /device/5 does not exist.
405 MethodNotAllowed	The request is not supported by the server.

- **Status codes 500 to 599 - Server Error**

Error	Reason
500 InternalServerError	The request was well-formed and valid, but the system encountered an issue while trying to execute the request. An example would be if the system was in the process of issuing a Create Device, and the physical device was unplugged before the add could complete. The system would report that the device could not be added.
503 ServiceUnavailable	This would only be encountered if the LEAP Server was in the process of shutting down.

Property	Required	Type	Details
Message	✓	string	A human readable description of what happened. This string should not be exposed to end customers, it is meant to help developers.
ErrorCode		int	In some cases, it is possible for a developer to write logic against an error and offer fixes to the end customer. When this is the case, ErrorCode will be returned so that code can be written against it. Possible error codes are documented along with the request that can generate it.
ErrorDetails		string	Details in addition to the error message. This string should not be exposed to end customers, it is meant to help developers.

Example exception response:

```
{
    "CommuniqueType": "ExceptionResponse",
    "Header": {
        "MessageBodyType": "ExceptionDetail",
        "StatusCode": "500 InternalServerError",
        "Url": "/device"
    },
    "Body": {
        "Message": "Device could not be added.",
        "ErrorCode": 5
    }
}
```

HyperReference Resource

When a resource definition contains references to other resources, hyper-references are returned. A hyper-reference contains only the URL (href) for the resource. Each resource in LEAP has a unique identifier called "href". For a given project, an href will never be reused. For instance, if a user adds a dimmer, it gets href /device/2, the user removes the dimmer, then adds it again - it will now get a different href, perhaps /device/3. Clients must not interpret the href.

When one resource refers to another, it will appear in a json struct with a single field: href.

Property	Required	Type	Details
href	✓	string	A unique identifier for a single resource

LEAP Object Model

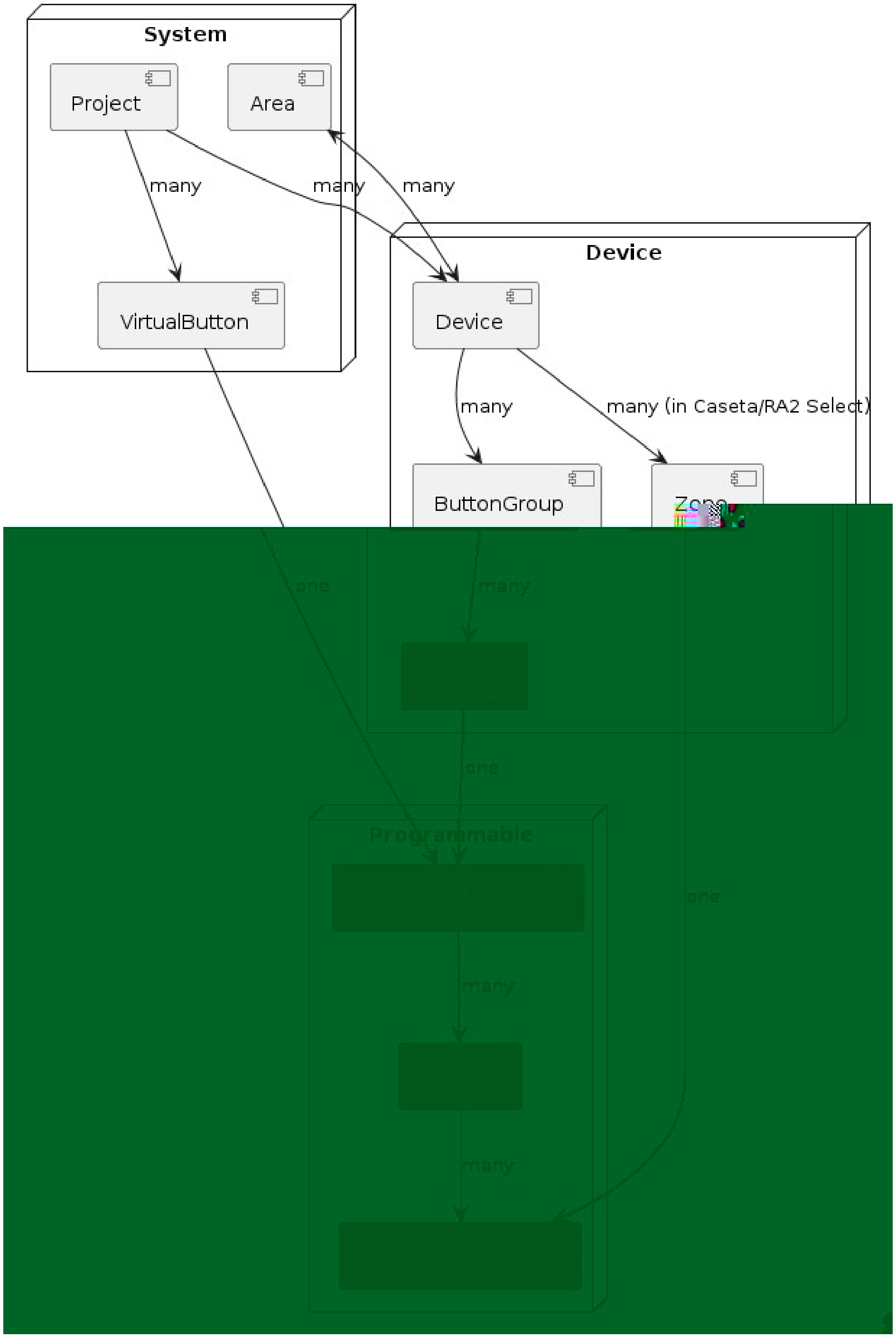
There are two fundamental relationships between two resources in the LEAP Object Model:

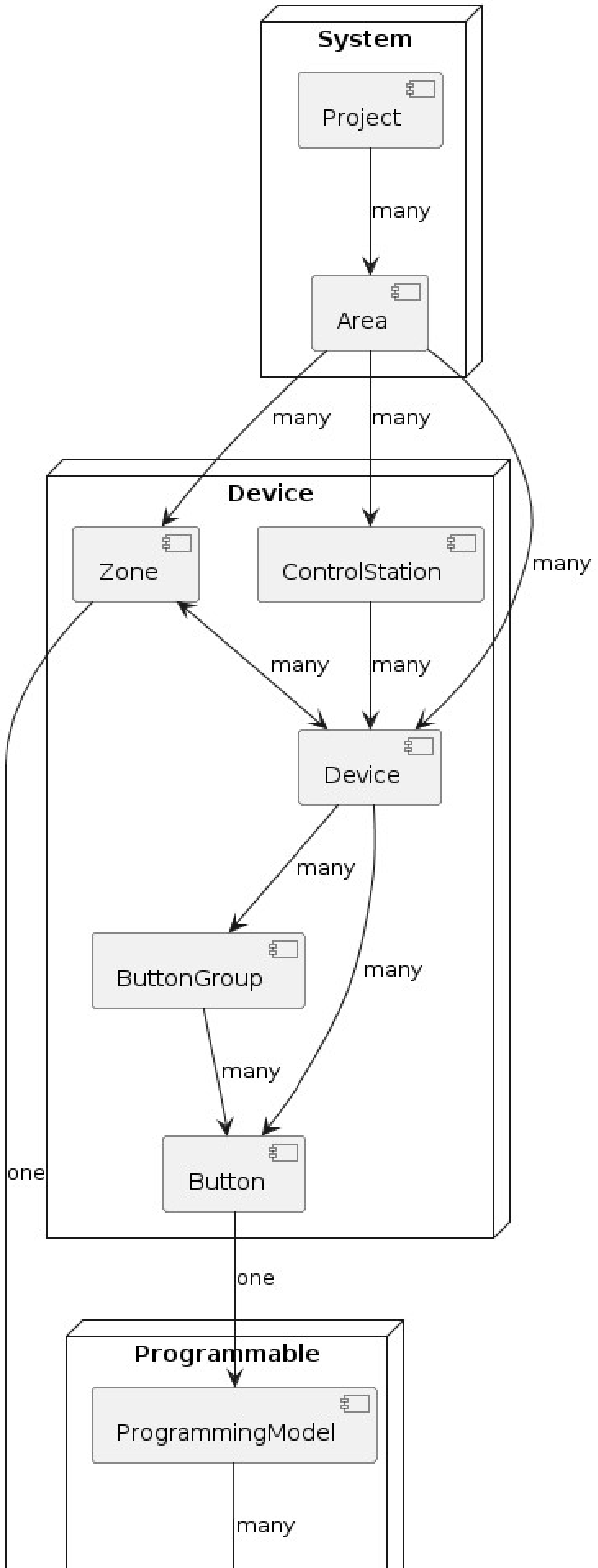
- **Parent-Child:** In a parent-child relationship, if the parent is deleted the child will also be deleted. One example is Devices, ButtonGroup, and Buttons. A physical button is the child of a buttongroup, which is a child of a device. So if the device is removed, so are its button groups, and then the buttons as well.
- **Association**: An association relationship is when two things are related, but one can be deleted and the other will still exist. For instance, a device can be associated with a room. However, deletion of the room or deletion of the device does not mean the other is deleted.

The following diagram describes the types of LEAP resources and how they relate to one another in different systems. See specific resources for more information about them.

Legend

Arrow-type	Relationship
→	Parent-Child
↔	Association


System
Project
Area
many
many
many
VirtualButton
Device
Device
many
many (in Caseta/RA2 Select)
ButtonGroup
Zone
one
many
one
Programmable
one
many
many


System
Project
many
Area
many
many
many
Device
Zone
ControlStation
many
many
many
Device
many
many
ButtonGroup
many
Button
one
one
Programmable
ProgrammingModel
many

Project

Caséta	✓
RA2 Select	✓
HomeWorks	✓
RadioRA 3	✓

Integrator Type	Supported
Cert-Based	✓
Username/Password	✗

Project Resource

Property	Required	Type	Details
href	✓	string	The unique identifier for this resource.
Name		string	The name of the project.
ProductType	✓	ProductType	The system type
MasterDeviceList	✓	DeviceReferences	A list of processors in the system. For Caséta/RA2 Select: There will be only one entry and it will be the bridge or main repeater itself. For HomeWorks/RadioRA 3: There will be an entry that corresponds to each proc in the system
ProjectModifiedTimestamp		Timestamp	The date project is modified. Only supported for **HomeWorks & RadioRA 3.**

ProductType Enum

Value	Description
Lutron Smart Bridge Project	A Caséta System

Lutron RA2 Select Project	A RA2 Select System
Lutron HWQS Project	A HomeWorks QSX system
Lutron RadioRA 3 Project	A RadioRA 3 system

DeviceReferences

Property	Required	Type	Details
Devices		List<HyperReference>	These HRefs can be used to request the details of the bridge or main repeater, such as the serial number.

Client Request

```
{
        "CommuniqueType": "ReadRequest",
        "Header": {
                "Url": "/project"
        }
}
```

Server Response

```
{
  "CommuniqueType": "ReadResponse",
  "Header": {
    "MessageBodyType": "OneProjectDefinition",
    "StatusCode": "200 OK",
    "Url": "/project"
  },
  "Body": {
    "Project": {
      "href": "/project",
      "Name": "SmartBridge",
      "ProductType": "Lutron Smart Bridge Project",
      "MasterDeviceList": {
        "Devices": [
          {
            "href": "/device/1"
          }
        ]
      },
      "ProjectModifiedTimestamp": {
        "Year": 2019,
        "Month": 7,
        "Day": 3,
        "Hour": 13,
        "Minute": 12,
        "Second": 35
      }
    }
  }
}
```

Areas

Caséta	✓	
RA2 Select	✓	
HomeWorks	⚠	List read on /area is not supported. Refer this to discover areas.
RadioRA 3	⚠	List read on /area is not supported. Refer this to discover areas.

Integrator Type	**Supported**	
Cert-Based	✓	⚠Read on /area is not supported for HomeWorks and RadioRA 3.
Username/Password	✓	⚠Read on /area is not supported.

Area Resource

The area resource describes a virtual grouping existing in the system. This is typically used to group devices, shade groups, or occupancy sensors into physical rooms.

Property	**Required**	**Type**	**Details**
href	✓	string	The unique identifier for this resource.
Name	✓	string	This is the "friendly" name of the area.
Parent	✓	HyperReference	The object owning this resource. When the parent is removed, this area will be removed as well.
AssociatedZones		List<HyperReference>	List of zones associated to this area.

AssociatedDevices		List<HyperReference>	**Caséta and RA2 Select Only.** Devices can be "grouped" into areas. This association is not required, and may be omitted.
AssociatedOccupancyGroups		List<HyperReference>	**Caséta and RA2 Select Only.** Occupancy Groups can be associated with an area. This association will determine the results of automatic programming in the Caséta and RA2 Select systems. To add or remove an occupancy group from an area, modify the occupancy group's AssociatedAreas property.

Area Reads

Client sends All Areas Definition read request

```
{
  "CommuniqueType":"ReadRequest",
  "Header":{
    "Url":"/area"
  }
}
```

Master Device sends Multiple Area Definition response

```
{
    "CommuniqueType": "ReadResponse",
    "Header": {
        "StatusCode": "200 OK",
        "Url": "/area",
        "MessageBodyType": "MultipleAreaDefinition"
    },
    "Body": {
        "Areas": [
            {   // The root area will always exist but is only used for
organizational purposes.
```

```
                              // All other areas have the root /area/1 as
their Parent.
                "href": "/area/1",
                "Name": "root"
            },
            {
                "href": "/area/2",
                "Name": "Kitchen",
                "Parent": {
                    "href": "/area/1"
                }
            },
                       // All other areas would be listed here as well.
        ]
    }
}
```

Area Subscribes

Client sends subscribe to all area configuration changes

Example Request

```
{
    "CommuniqueType": "SubscribeRequest",
    "Header": {
        "Url": "/area",
        "Directives": {
            "SuppressMessageBody": true
        }
    }
}
```

Example Response

```
{
    "CommuniqueType": "SubscribeResponse",
    "Header": {
        "StatusCode": "204 NoContent",
        "Url": "/area",
        "Directives": {
            "SuppressMessageBody": true
        }
    }
}
```

Example Asynchronous Response

```
{
    "CommuniqueType": "ReadResponse",
    "Header": {
        "Url": "/area",
        "StatusCode": "200 OK",
        "MessageBodyType": "MultipleAreaDefinition"
    },
    "Body": {
        "Areas": [
            {
                "href":"/area/#",
                "XID":"xyz",
                "Name":"Area 001",
                "Parent":{
                    "href":"/area/#"
                },
                "AssociatedZones":[
                    {
                        "href":"/zone/#"
                    }
                ],
                "AssociatedControlStations":[
                    {
                        "href":"/controlstation/#"
                    }
                ]
            }
        ]
    }
}
```

Area Status Resource

Caséta	✗
RA2 Select	✗
HomeWorks	✓
RadioRA 3	✓

Integrator Type	Supported
Cert-Based	✓
Username/Password	✓

An area status is used to get runtime information on the state of the area, example, occupancy status of the area.

Property	Required	Type	Details
href	✓	string	The unique identifier for this resource.
Area	✓	HyperReference	The area with which this status is associated.
OccupancyStatus	✗	OccupancyStatus	The current occupancy status of the area.

CurrentScene	✖	HyperReference	The current active scene on the area. If there isn't an active scene, then CurrentScene will be set to null.
Level	✖	int	The maximum level of the area's lighting zones.

Occupancy Status Enum

Value	Description
Occupied	At least one occupancy group in the area is occupied.
Unoccupied	None of the occupancy group in the area are reporting occupied.
Unknown	None of the occupancy groups are occupied, and not all of the other occupancy groups are unoccupied. This means we cannot accurately determine area is Unoccupied, so the status is instead Unknown.

Area Status Reads

Client sends one Area Read Request for Status:

```
{
  "CommuniqueType":"ReadRequest",
  "Header":{
    "Url":"/area/23/status"
  }
}
```

Master Device sends Read Response:

```
{
        "CommuniqueType": "ReadResponse",
        "Header": {
                "MessageBodyType": "OneAreaStatus",
                "StatusCode": "200 OK",
                "Url": "/area/23/status"
        },
        "Body": {
                "AreaStatus":
         {
            "href" : "/area/23/status"
            "OccupancyStatus":"Occupied",
                       "CurrentScene" : {
                               "href": "/myscene",
                       }
                }
        }
}
```

Client sends All Area Read Request for Status:

```
{
  "CommuniqueType":"ReadRequest",
  "Header":{
    "Url":"/area/status"
  }
}
```

Master Device sends Read Response:

```
{
        "CommuniqueType": "ReadResponse",
        "Header": {
                "MessageBodyType": "MultipleAreaStatus",
                "StatusCode": "200 OK",
                "Url": "/area/status"
        },
        "Body": {
                "AreaStatuses":[
         {
            "href" : "/area/23/status"
            "OccupancyStatus":"Occupied",
                        "CurrentScene" : {
                                "href": "/myscene",
                        }
                },
        {
            "href" : "/area/499/status"
            "OccupancyStatus":"Unknown"
                        "CurrentScene" : {
                                "href": "/anotherscene",
                        }
                }
        ]
        }
}
```

Area Status Subscription

Client sends All Area Subscription Request for Status:

```
{
    "CommuniqueType": "SubscribeRequest",
    "Header": {
        "Url": "/area/status"
    },
}
```

Master Device sends Successful subscription Response:

```
{
    "CommuniqueType": "SubscribeResponse",
    "Header": {
        "StatusCode": "204 NoContent",
        "Url": "area/status"
}
```

Command an Area to a Level

GoToGroupLightingLevel

This request will only work to control all the lighting zones in an area.

Integrator Type	Supported
Cert-based	✓
Username/Password	✓

Property	Required	Type	Details
Level	✓	int	• Obeyed by all lighting zones • Switched zones will do the logical thing based on raise/lower configuration
VibrancyStatus		VibrancyStatus	Ignored by all zones that do not have an AvailableControlType of Vibrancy.
ColorTuningStatus		ColorTuningStatus	• The fields inside are mutually exclusive. • Ignored by zones that do not have the requisite AvailableControlType. • Sending CurveDimming (i.e., enabling warm dim) is not supported.
DelayTime		Timespan	Specifies a delay prior to starting the fade.
FadeTime		Timespan	• Specifies the amount of fade time from the current level. • Ignored by zones that cannot fade (e.g., switched lighting)

VibrancyStatus

Property	Type	Details
Vibrancy	int	- Must **not** be specified if AutoVibrancy is specified
AutoVibrancy	EnabledType	- Must **not** be specified if Vibrancy is specified

Client sends a request to control all the lighting zones in an area

```
{
    "CommuniqueType":"CreateRequest",
    "Header":{
        "Url":"<area-href>/commandprocessor"
    }
    "Body: {
               "Command": {
            "CommandType": "GoToGroupLightingLevel",
                "GroupLightingLevelParameters": {
                       "Level": 100,
                               "VibrancyStatus": {
                          "Vibrancy": "100",
                          "AutoVibrancy": "Enabled"
                       }
                               "ColorTuningStatus": {
                      "XYTuningLevel": {"X": 0, "Y": 0},
                   "HSVTuningLevel": {"Hue": 100, "Saturation": 100},
                  "WhiteTuningLevel": {"Kelvin": 10000}
                   },
                "FadeTime": "5",
                    "DelayTime": "1"
                }
               }
    }
}
```

Server responds that the request was successfully executed

```
{
    "CommuniqueType":"CreateResponse",
    "Header":{
        "Url":"/area/1/commandprocessor",
        "StatusCode":"204 NoContent"
    }
}
```

Command an Area to activate scene

GoToScene

This request will only work to activate scene in an area.

Integrator Type	Supported
Cert-based	✓
Username/Password	✓

Property	Required	Type	Details
CurrentScene	✓	HyperReference	HRef of the scene that should be activated for an area.

Client sends a request to activate a scene in an area.

```
{
    "CommuniqueType": "CreateRequest",
    "Header": {
        "Url": "/area/26/commandprocessor"
    },
    "Body": {
        "Command": {
            "CommandType": "GoToScene",
            "GoToSceneParameters": {
                "CurrentScene": {
                    "href": "/areascene/30"
                }
            }
        }
    }
}
```

Server responds that the request was successfully executed

```
{
    "CommuniqueType":"CreateResponse",
    "Header":{
        "Url":"/area/26/commandprocessor",
        "StatusCode":"204 NoContent"
    }
}
```

Area Scene Resource

The Area Scene resource contains information about pre-defined programming for an area.

Integrator Type	Supported
Cert-based	✓
Username/Password	✓

Property	Required	Type	Details
href	✓	string	The unique identifier for this resource.
Name	✓	string	This is the "friendly" name of this resource
Parent	✓	HyperReference	The object owning this resource. When the parent is removed, this AreaScene will be removed as well.
Preset	✓	HyperReference	HRef to the Preset object. A read on the Preset would provide more information about the specific programming on the AreaScene. See Residential Systems LEAP Protocol - Control and Monitoring#ControlandMonitoring-Presets for more information.

Area Scene Reads

Client reads the AreaScenes for a specific area

```
{
  "CommuniqueType":"ReadRequest",
  "Header":{
    "Url":"<area-href>/areascene"
  }
}
```

Server responds with the list of AreaScenes

```
{
  "CommuniqueType": "ReadResponse",
  "Header": {
    "MessageBodyType": "MultipleAreaSceneDefinition",
    "StatusCode": "200 OK",
    "Url": "<area-href>/areascene"
  },
  "Body": {
    "AreaScenes": [
      {
        "href": "/areascene/566",
        "Name": "Off Scene",
        "Parent": {
          "href": "/area/562"
        },
        "Preset": {
          "href": "/preset/566"
        }
      },
      {
        "href": "/areascene/567",
        "Name": "Scene 001",
        "Parent": {
          "href": "/area/562"
        },
        "Preset": {
          "href": "/preset/567"
        }
      },
      {
        "href": "/areascene/568",
        "Name": "Scene 002",
        "Parent": {
          "href": "/area/562"
        },
        "Preset": {
          "href": "/preset/568"
        }
      },
      {
        "href": "/areascene/569",
        "Name": "Scene 003",
        "Parent": {
          "href": "/area/562"
        },
        "Preset": {
          "href": "/preset/569"
        }
      }
    ]
  }
}
```

To discover the root area in HomeWorks/RadioRA 3 system, read on /area/rootarea

Integrator Type	**Supported**
Cert-Based	✓

Integrator Type	Supported
Username/Password	✓

```
{
  "CommuniqueType": "ReadRequest",
  "Header": {
    "Url": "/area/rootarea"
  }
}
{
  "CommuniqueType": "ReadResponse",
  "Header": {
    "MessageBodyType": "OneAreaDefinition",
    "StatusCode": "200 OK",
    "Url": "/area/rootarea"
  },
  "Body": {
    "Area": {
      "href": "/area/3",
      "Name": "HomeWorks Project"
    }
  }
}
```

Discover all the child areas by read on href /area/id/childarea/summary

Child Area Summary Resource

Property	Required	Type	Details
href	✓	string	The unique identifier for this particular child area.
Name	✓	string	This is the "friendly" name of the area.
SortOrder		int	The sort order of the area
IsLeaf		bool	Flag specifying whether the area is considered a leaf area. IsLeaf = true means that the area has no children
Parent		HyperReference	The HyperReference of the area's parent area

Integrator Type	Supported
Cert-Based	✓
Username/Password	✓

```
{
  "CommuniqueType": "ReadRequest",
  "Header": {
    "Url": "/area/3/childarea/summary"
  }
}
{
  "CommuniqueType": "ReadResponse",
  "Header": {
    "MessageBodyType": "MultipleAreaSummaryDefinition",
    "StatusCode": "200 OK",
    "Url": "/area/3/childarea/summary"
  },
  "Body": {
    "AreaSummaries": [
      {
        "href": "/area/71",
        "Name": "Area",
        "Parent": {
          "href": "/area/3"
        },
        "SortOrder": 0,
        "IsLeaf": true
      },
      {
        "href": "/area/757",
        "Name": "Another area",
        "Parent": {
          "href": "/area/3"
        },
        "SortOrder": 1,
        "IsLeaf": true
      }
    ]
  }
}
```

The IsLeaf property is set to true to indicate that the area doesnt have any more child areas.

Client sends One Area Definition read request

```
{
  "CommuniqueType":"ReadRequest",
  "Header":{
    "Url":"/area/2"
  }
}
```

Master Device sends One Area Definition response

```
{
    "CommuniqueType": "ReadResponse",
    "Header": {
        "StatusCode": "200 OK",
        "Url": "/area/2",
        "MessageBodyType": "OneAreaDefinition"
    },
    "Body": {
        "Area": {
            "href": "/area/2",
            "Name": "Kitchen",
            "Parent": {
                "href": "/area/1"
            }
               }
        // Can contain additional product specific fields.
    }
}
```

Area Association Resource

Property	Required	Type	Details
href	✓	string	The unique identifier for this particular association.
Area	✓	HyperReference	The owner of the AreaAssociation resource is associated with the area referenced here.

Control Station

Caséta	
RA2 Select	
HomeWorks	
RadioRA 3	

Integrator Type	Supported	
Cert-Based		
Username/Password		Read on /controlstation/id and /area/id/associatedcontrolstation is supported.

Control Station Resource

Property	Required	Type	Details
href		string	The unique identifier for this particular control station.
Name		string	This is the "friendly" name of the control station.
AssociatedArea		HyperReference	The area href that this control station is associated with

SortOrder		int	The sort order for this control station
AssociatedGangedDevices		List<AssociatedGangedDevice>	The list of associated ganged devices for the control station

AssociatedGangedDevice

Property	Required	Type	Details
Device	✓	Device	
GangPosition	✓	int	It is the position of the device inside of a control station.

Control Station Reads

Discovering control stations

```
{
  "CommuniqueType": "ReadRequest",
  "Header": {
    "Url": "/area/757/associatedcontrolstation"
  }
}
{
  "CommuniqueType": "ReadResponse",
  "Header": {
    "MessageBodyType": "MultipleControlStationDefinition",
    "StatusCode": "200 OK",
    "Url": "/area/757/associatedcontrolstation"
  },
  "Body": {
    "ControlStations": [
      {
        "href": "/controlstation/779",
        "Name": "SeeTouch Keypad",
        "AssociatedArea": {
          "href": "/area/757"
        },
        "SortOrder": 0,
        "AssociatedGangedDevices": [
          {
            "Device": {
              "href": "/device/781",
              "DeviceType": "SeeTouchKeypad"
            },
            "GangPosition": 0
          }
        ]
      },
      {
        "href": "/controlstation/857",
        "Name": "Maestro Dimmer",
        "AssociatedArea": {
          "href": "/area/757"
        },
        "SortOrder": 2,
        "AssociatedGangedDevices": [
          {
            "Device": {
              "href": "/device/859",
              "DeviceType": "Unknown"
            },
            "GangPosition": 0
          }
        ]
      }
    ]
  }
}
```

Control Station Subscribes

Subscribe to configuration changes on all control stations

Example Request

```
{
    "CommuniqueType": "SubscribeRequest",
    "Header": {
        "Url": "/controlstation",
        "Directives": {
            "SuppressMessageBody": true
        }
    }
}
```

Example Response

```
{
    "CommuniqueType": "SubscribeResponse",
    "Header": {
        "StatusCode": "204 NoContent",
        "Url": "/controlstation",
        "Directives": {
            "SuppressMessageBody": true
        }
    }
}
```

Example Asynchronous Response

```
{
    "CommuniqueType": "ReadResponse",
    "Header": {
        "Url": "/controlstation",
        "StatusCode": "200 OK",
        "MessageBodyType": "MultipleControlStationDefinition"
    },
    "Body": {
        "ControlStations": [
            {
                "href":"/controlstation/#",
                "Name":"Control Station 001",
                "AssociatedArea":{
                    "href":"/area/#"
                },
                "AssociatedGangedDevices":[
                    {
                        "Device":{
                            "DeviceType":"Unknown",
                            "href":"/device/#",
                            "AddressedState":"Addressed"
                        },
                        "GangPosition":2
                    }
                ],
                "SortOrder":0
            }
        ]
    }
}
```

Device

Caséta	✓	
RA2 Select	✓	
HomeWorks	⚠	List reads on device will not be supported. Client has to do device/id read on individual devices to get the data.
RadioRA 3	⚠	List reads on device will not be supported. Client has to do device/id read on individual devices to get the data.

Integrator Type	**Supported**	
Cert-Based	✓	⚠Read on /device is not supported for HomeWorks/RadioRA 3.
Username/Password	✓	⚠Read on /device is not supported.

The device resource describes a physical device with a serial number existing in the system. There are two special cases in the Connect systems:

Device Resource

- **Homeworks Homeowner Keypads**: These are virtual keypads that will appear as device resources without serial numbers. This is exposed as a device because it shows up in the homeowner's Lutron Connect App.
- **Homeworks Phantom Keypads**: These are virtual keypads that will not show up as devices since they don't show up in the homeowner's Lutron Connect App. They exist exclusively for use by third party Lutron Integration Protocol (LIP) integrators.

Property	Required	Type	Details
href	✓	string	The unique identifier for this resource.
Name	✓	string	This is the "friendly" name of the device.
Parent		HyperReference	The object owning this resource. Typically, this will be the project.
ModelNumber		string	The model number of the device. Any unknown values (such as color code) will be replaced with the character X. e.g. A Caséta dimmer with model number PD-6WCL-WH-R may appear here as PD-6WCL-XX-R since we do not know the color of the dimmer. **Only in Caséta/RA2 Select**
AssociatedArea		HyperReference	Devices can be "grouped" into areas. This association is not required, and may be omitted.

LocalZones		List<HyperReference>	List of zones connected to this device via a switchleg. In the connect system, please issue a read on /zone to simply get a list of all zones in the system. In HomeWorks/RadioRA 3, please issue a read on /area/id/zone to get a list of all zones in the system
FullyQualifiedName	✓	List<string>	Includes the area hierarchy and device name that, when concatenated, becomes the unambiguous name for this device. e.g. Assume the "First Floor" (/area/2) contains an area called "Kitchen" (/area/5), and also assume /device/2 ("Ceiling Light") is associated with /area/5. The FullyQualifiedName would be: `["First Floor","Kitchen","Ceiling Light"]` **`Only in Caséta/RA2 Select.`**
SerialNumber		int	The serial number of this device. Ranges from 0-4294967295. This will be omitted if this device is not activated or is a homeowner keypad.
DeviceType	✓	DeviceType	The category of the device, for the sake of displaying an image to an end user. **Only in Caséta, RA2 Select, HomeWorks and RadioRA 3**
EngravingKit		EngravingKit	Some products are manufactured with a standard set of engravings. This property will be present for those products.

ButtonGroups		List<HyperReference>	If the device contains buttons (e.g. Pico) then this property will be present and contain the HyperReferences to the buttongroups for this device. A buttongroup represents a collection of buttons acting together. It is possible for a single device to contain multiple button groups. For instance, the Lutron 4-Group Remote has 5 button groups (one for each of the four groups on the remote, and one for the "All" group). There is also a four button pico that has two groups (two sets of two buttons).
OccupancySensors		List<HyperReference>	If the device includes occupancy sensors, then this property will be present and contain the HyperReferences to the OccupancySensors attached to this device. A wall sensor will only ever have one occupancy sensor attached. A product like the ESN may have up to 4 sensors attached. **Only in RA2 Select**
FirmwareImage		FirmwareImage	Includes details about the firmware currently installed on the device. If omitted, the firmware details are not known.

Engraving Kit Enum

Engraving kit enum is only supported for Pico Wireless Control.

Value	Details
HomeAway	Describes https://www.lutron.com/TechnicalDocumentLibrary/3691066_ENG.pdf, Entry - P01

Bedside	Describes https://www.lutron.com/TechnicalDocumentLibrary/3691066_ENG.pdf, Bedside - P02
Kitchen	Describes https://www.lutron.com/TechnicalDocumentLibrary/3691066_ENG.pdf, Kitchen - P02
LivingRoom	Describes https://www.lutron.com/TechnicalDocumentLibrary/3691066_ENG.pdf, Family Room - P01
AnyRoomRelax	Describes https://www.lutron.com/TechnicalDocumentLibrary/3691066_ENG.pdf, Any Room - P03
Lights	Describes http://www.lutron.com/TechnicalDocumentLibrary/369612.pdf Light (L01) Note: Only newer picos will report this value. There are many picos shipping with this engraving that do not report any engraving kit value. For those, you may want to ask the end user what kind of pico they have and use their response to show a picture of the device.
Shades	Describes http://www.lutron.com/TechnicalDocumentLibrary/369612.pdf Shade (icons) (S01) Note: Only newer picos will report this value. There are many picos shipping with this engraving that do not report any engraving kit value. For those, you may want to ask the end user what kind of pico they have and use their response to show a picture of the device.
Fans	Describes http://www.lutron.com/TechnicalDocumentLibrary/369612.pdf Fans (icons)
Audio	Describes http://www.lutron.com/TechnicalDocumentLibrary/3691021.pdf Note: Only newer picos will report this value. There are many picos shipping with this engraving that do not report any engraving kit value. For those, you may want to ask the end user what kind of pico they have and use their response to show a picture of the device.
HorizontalSheerBlinds	Describes http://www.lutron.com/TechnicalDocumentLibrary/369612.pdf, Horizontal Sheer Blinds Model: S09

Custom	Describes https://www.lutron.com/TechnicalDocumentLibrary/3691066_ENG.pdf with a custom engraving. Note: Only newer picos will report this value. There are many picos shipping with this engraving that do not report any engraving kit value. For those, you may want to ask the end user what kind of pico they have and use their response to show a picture of the device.

Firmware Image Resource

Integrator Type	Supported
Cert-Based	✓
Username/Password	✗

Property	Required	Type	Details
Firmware	✓	Firmware	Details about the firmware itself, such as version and build date.
Installed		Timestamp	The date the firmware was installed on this device.

Firmware Resource

Property	Required	Type	Details
DisplayName	✓	string	A human readable form of the firmware for display purposes only. The format is device specific.

Device Reads

Client sends all Device Definition Read Request:

Returns a list of all devices in the system. New systems will always have at least one device, the master device.

```
{
  "CommuniqueType":"ReadRequest",
  "Header":{
    "Url":"/device"
  }
}
```

Master Device sends Read Response:

```
{
  "CommuniqueType":"ReadResponse",
  "Header":{
    "MessageBodyType":"MultipleDeviceDefinition",
    "StatusCode":"200 OK",
    "Url":"/device"
  },
  "Body":{
    "Devices":[
      {
        "Name":"Smart Bridge",
        "Parent":{
          "href":"/project"
        },
        "href":"/device/1",
        "SerialNumber":8051960,
        "DeviceType":"SmartBridge",
        "FullyQualifiedName":[
          "Smart Bridge"
        ],
        "FirmwareImage":{
                "Firmware":{
                "DisplayName":"01.07.01a000",
                },
                "Installed":{
                "Year":2018,
                ...,
                "Utc":"0"
                }
                }
      },
      {
        "Name":"Light",
        "Parent":{
          "href":"/project"
        },
        "href":"/device/4",
        "SerialNumber":7140406,
        "DeviceType":"WallDimmer",
        "LocalZones":[
          {
            "href":"/zone/2"
          }
        ],
        "FullyQualifiedName":[
          "Kitchen",
          "Light"
        ],
        "AssociatedArea":{
          "href":"/area/4"
        }
      },
      {
        "Name":"Main Pico",
        "Parent":{
          "href":"/project"
        },
        "href":"/device/5",
        "SerialNumber":10778088,
        "DeviceType":"Pico3ButtonRaiseLower",
        "ButtonGroups":[
          {
            "href":"/buttongroup/2"
          }
        ],
        "FullyQualifiedName":[
          "Main Pico"
               ],
      },
      {
        "Name":"Occupancy Sensor",
        "Parent":{
          "href":"/project"
        },
        "AssociatedArea":{
          "href":"/area/4"
        }
        "href":"/device/6",
        "SerialNumber":15438088,
        "DeviceType":"RPSOccupancySensor",
        "OccupancySensors":[
          {
            "href":"/occupancysensor/2"
          }
        ],
        "FullyQualifiedName":[
            "Kitchen",
            "Occupancy Sensor"
               ],
      }
    ]
  }
}
```

Client sends one Device Definition Read Request:

```
{
  "CommuniqueType":"ReadRequest",
  "Header":{
    "Url":"/device/1"
  }
}
```

Master Device sends Read Response:

```
{
  "CommuniqueType":"ReadResponse",
  "Header":{
    "MessageBodyType":"OneDeviceDefinition",
    "StatusCode":"200 OK",
    "Url":"/device/1"
  },
  "Body":{
    "Device":{
      "Name":"Smart Bridge",
      "Parent":{
        "href":"/project"
      },
      "href":"/device/1",
      "SerialNumber":8051960,
      "DeviceType":"SmartBridge",
      "FullyQualifiedName":[
        "Smart Bridge"
      ]
    }
  }
}
```

TemperatureSensors

Caséta	✗	
RA2 Select	✗	
HomeWorks	✓	List reads are not supported. Client is expected to get temperature sensors for each device.
RadioRA 3	✗	

Integrator Type	Supported
Cert-Based	✓
Username/Password	✓

TemperatureSensor Reads

A temperature sensor refers to a physical temperature sensor present on certain devices.

Property	Required	Type	Details
href	✓	HyperReference	The unique identifier for this resource.
Parent	✓	HyperReference	A reference to the device that owns this temperature sensor.

Client sends TemperatureSensor Read Request:

```
{
        "CommuniqueType": "ReadRequest",
        "Header": {
                "Url": "/temperaturesensor/2"
        }
}
```

Master Device sends Read Response:

```
{
        "CommuniqueType": "ReadResponse",
        "Header": {
                "MessageBodyType": "OneTemperatureSensorDefinition",
                "StatusCode": "200 OK",
                "Url": "/temperaturesensor/2"
        },
        "Body": {
                "TemperatureSensor": {
                        "href": "/temperaturesensor/2",
                        "Parent": {
                                "href": "/device/7"
                        }
                }
        }
}
```

Client sends Device Temperature Sensor Read Request:

```
{
  "CommuniqueType": "ReadRequest",
  "Header": {
    "Url": "/device/573/temperaturesensor"
  }
}
```

Master Device sends Read Response:

```
{
  "CommuniqueType": "ReadResponse",
  "Header": {
    "MessageBodyType": "MultipleTemperatureSensorDefinition",
    "StatusCode": "200 OK",
    "Url": "/device/573/temperaturesensor"
  },
  "Body": {
    "TemperatureSensors": [
      {
        "href": "/temperaturesensor/753",
        "Parent": {
          "href": "/device/573"
        }
      },
      {
        "href": "/temperaturesensor/986",
        "Parent": {
          "href": "/device/573"
        }
      }
    ]
  }
}
```

TemperatureSensor Status Reads

A temperature sensor refers to a physical temperature sensor present on certain devices.

Property	Required	Type	Details
href	✓	HyperReference	The unique identifier for this resource.
TemperatureSensor	✓	HyperReference	A reference to the the temperature sensor.
CurrentTemperature	✓	Setpoint	The temperature on the Fahrenheit and the Celsius scale

Client sends TemperatureSensor Status Read Request:

```
{
        "CommuniqueType": "ReadRequest",
        "Header": {
                "Url": "/temperaturesensor/2/status"
        }
}
```

Master Device sends Read Response:

```
{
        "CommuniqueType": "ReadResponse",
        "Header": {
                "MessageBodyType": "OneTemperatureSensorStatus",
                "StatusCode": "200 OK",
                "Url": "/temperaturesensor/2/status"
        },
        "Body": {
                "TemperatureSensor": {
                        "href": "/temperaturesensor/2/status",
                        "TemperatureSensor": {
                                "href": "/temperaturesensor/2"
                        },
            "CurrentTemperature": {
                "C": 0,
                    "F": 32
                }
                    }
          }
}
```

ButtonGroup

Caséta	✓	
RA2 Select	✓	
HomeWorks	⚠	Read on /buttongroup is not supported.
RadioRA 3	⚠	Read on /buttongroup is not supported.

Integrator Type	Supported
Cert-Based	✓
Username/Password	✓

ButtonGroup Resource

A button group is a logical collection of buttons on a device. For instance, a 2 button pico has a single buttongroup with two buttons. A 2 group, 2 button pico (having two sets of on-off buttons) has two button groups, each with two buttons.

Property	Required	Type	Details
href	✓	HyperReference	The unique identifier for this resource.
Buttons	✓	List<HyperReference>	Hrefs to the member buttons of this buttongroup
AffectedZones		List<HyperReference>	Column Programming Only (i.e., on a pico that is not button-by-button programmed). This field contains a collection of zones controlled by this device.

StopIfMoving		EnabledType	Column Programming Only (i.e., on a pico that is not button-by-button programmed). When enabled, if any shade zones appear in AffectedZone, if they are moving when a button is pressed then they will stop moving rather than going to the new commanded level. If Disabled, shades will always go to the commanded level when a button is pushed.
Parent	✓	HyperReference	A reference to the device that owns this button group.
SortOrder		int	The sort order of the buttongroup in a list read.
Category		ButtonGroupCategory	Category of this button group.
ProgrammingType		ProgrammingType	Programming type of button group.

ButtongGroupCategory

Property	**Required**	**Type**	**Details**
Type	✓	string	The category of the button group. Values can be **"Lights"**, **"Shades"**, **"Audio"**, **"Fans"**, **"Unknown"**.

ProgrammingType Enum

Property	**Required**	**Type**	**Details**
Type	✓	string	The category of the button group. Values can be **"Column"** and **"Freeform"**.

Client sends Button Group Definition Read Request:

```
{
        "CommuniqueType": "ReadRequest",
        "Header": {
                "Url": "/buttongroup/2"
        }
}
```

Master Device sends Read Response:

```
{
        "CommuniqueType": "ReadResponse",
        "Header": {
                "MessageBodyType": "OneButtonGroupDefinition",
                "StatusCode": "200 OK",
                "Url": "/buttongroup/2"
        },
        "Body": {
                "ButtonGroup": {
                        "href": "/buttongroup/2",
                        "Buttons": [{
                                "href": "/button/51"
                        },
                        {
                                "href": "/button/52"
                        },
                        {
                                "href": "/button/53"
                        },
                        {
                                "href": "/button/54"
                        },
                        {
                                "href": "/button/55"
                        }],
                        "StopIfMoving": "Disabled",
                        "AffectedZones": [{
                                "href": "/buttongroup/2/affectedzone/6",
                                "Zone": {
                                        "href": "/zone/6"
                                }
                        },
                        {
                                "href": "/buttongroup/2/affectedzone/2",
                                "Zone": {
                                        "href": "/zone/2"
                                }
                        },
                        {
                                "href": "/buttongroup/2/affectedzone/3",
                                "Zone": {
                                        "href": "/zone/3"
                                }
                        }],
                        "Parent": {
                                "href": "/device/7"
                        }
                }
        }
}
```

Client sends Device Button Group Read Request:

```
{
  "CommuniqueType": "ReadRequest",
  "Header": {
    "Url": "/device/573/buttongroup"
  }
}
```

Master Device sends Read Response:

```
{
  "CommuniqueType": "ReadResponse",
  "Header": {
    "MessageBodyType": "MultipleButtonGroupDefinition",
    "StatusCode": "200 OK",
    "Url": "/device/573/buttongroup"
  },
  "Body": {
    "ButtonGroups": [
      {
        "href": "/buttongroup/663",
        "Buttons": [
          {
            "href": "/button/664"
          },
          {
            "href": "/button/666"
          },
          {
            "href": "/button/668"
          },
          {
            "href": "/button/670"
          },
          {
            "href": "/button/672"
          },
          {
            "href": "/button/674"
          },
          {
            "href": "/button/676"
          },
          {
            "href": "/button/678"
          },
          {
            "href": "/button/680"
          },
          {
            "href": "/button/682"
          },
          {
            "href": "/button/683"
          },
          {
            "href": "/button/684"
          },
          {
            "href": "/button/685"
          },
          {
            "href": "/button/686"
          },
          {
            "href": "/button/687"
          },
          {
            "href": "/button/688"
          },
          {
            "href": "/button/689"
          },
          {
            "href": "/button/690"
          },
          {
            "href": "/button/692"
          },
          {
            "href": "/button/694"
          },
          {
            "href": "/button/696"
          },
          {
            "href": "/button/698"
          },
          {
            "href": "/button/700"
          }
        ],
        "Parent": {
          "href": "/device/573"
        },
        "SortOrder": 0,
        "StopIfMoving": "Disabled",
        "ProgrammingType": "Freeform"
      },
      {
        "href": "/buttongroup/753",
        "Buttons": [
          {
            "href": "/button/754"
          },
          {
            "href": "/button/758"
          },
          {
            "href": "/button/762"
          },
          {
            "href": "/button/766"
          },
          {
            "href": "/button/770"
          }
        ],
        "Parent": {
          "href": "/device/573"
        },
        "SortOrder": 4,
        "StopIfMoving": "Disabled",
        "ProgrammingType": "Freeform"
      }
    ]
  }
}
```

Buttons

Caséta	✓	
RA2Select	✓	
HomeWorks	✓	⚠Read on /button is not supported. Read this to discover buttons.
RadioRA 3	✓	⚠Read on /button is not supported. Read this to discover buttons.

Integrator Type	Supported	
Cert-Based	✓	⚠Read on /button is not supported for HomeWorks and RadioRA 3
Username/Password	✓	⚠Read on /button is not supported.

Button Resource

A button corresponds with a physical button on a physical keypad that has been added to the system. LEAP clients can read metadata, and execute button presses.

Property	Required	Type	Details
href	✓	string	The unique identifier for this resource.
ButtonNumber	✓	int	A number to uniquely identify a button on a device. For clients that are trying to visually representing Lutron keypads in their user interfaces this number will allow you to place the button in the correct place.

ProgrammingModel		HyperReference	An href to a programming model object to give information about how the programming is structured for this button.
Parent	✓	HyperReference	An href to the parent buttongroup. This field will be omitted if no Parent exists.
Engraving		Engraving	The text engraved on the button.
Name		string	This is the "friendly" name of the button.
AssociatedLED		HyperReference	The LED associated with the button. This field will be omitted if no LED exists for the button

Engraving

Property	Required	Type	Details
Text	✓	string	The text engraved on the button.

Button Reads

Button Read Request

```
{
    "CommuniqueType": "ReadRequest",
    "Header": {
        "Url": "/button/2"
    }
}
```

Button Read Response

```
{
    "CommuniqueType": "ReadResponse",
    "Header": {
        "StatusCode": "200 OK",
        "Url": "/button/2",
        "MessageBodyType": "OneButtonDefinition"
    },
    "Body": {
        "Button": {
            "href": "button/2",
            "ButtonNumber": 3,
            "ProgrammingModel": {
                "href": "/programmingmodel/4"
            },
            "Parent": {
                "href": "/buttongroup/3"
            },
               "AssociatedLED": {
                       "href": "/led/789"
               }
        }
        }
}
```

Button Group Expanded

Caséta	✓
RA2Select	✓
HomeWorks	✓
RadioRA 3	✓

Integrator Type	Supported
Cert-Based	✓
Username/Password	✓

ButtonGroupCategoryEnum

Values
Lights
Shades
Audio
Fans
Appliances
HorizontalSheerBlinds
Unknown

ProgrammingType enum

Values
Column
FreeForm

AffectedZoneSummary

A description

Property	Required	Type	Details

href	✓	string	
Zone	✓	HyperReference	

ButtonGroupCategory

A description

Property	Required	Type	Details
Type	✓	ButtonGroupCategoryEnum	Type of the Button Group Category

ButtonGroup Resource

A description

Property	Required	Type	Details
href	✓	string	The unique identifier for this resource.
Buttons	✓	[]HyperReference	A number to uniquely identify a button on a device. For clients that are trying to visually representing Lutron keypads in their user interfaces this number will allow you to place the button in the correct place.
Affected Zones		[]AffectedZoneSummary	
Parent	✓	HyperReference	An href to the parent buttongroup. This field will be omitted if no Parent exists.
SortOrder		int	SortOrder in which the buttons are on the device
StopIfMoving		EnabledState	
Category		ButtonGroupCategory	The type of devices this button group is going to affect.
ProgrammingType		ProgrammingType	ProgrammingType of the buttongroup.

ButtonGroupExpanded Resource

A description

Property	Required	Type	Details
ButtonGroup	✓	ButtonGroup	The unique identifier for this resource.
Buttons	✓	[]*Button	A number to uniquely identify a button on a device. For clients that are trying to visually representing Lutron keypads in their user interfaces this number will allow you to place the button in the correct place.

ButtonGroupExpanded Read

Button Read Request

```
{
    "CommuniqueType": "ReadRequest",
    "Header": {
        "Url": "/device/2/buttongroup/expanded"
    }
}
```

Button Read Response

```
{
  "CommuniqueType": "ReadResponse",
  "Header": {
    "MessageBodyType": "MultipleButtonGroupExpandedDefinition",
    "StatusCode": "200 OK",
    "Url": "/device/781/buttongroup/expanded"
  },
  "Body": {
    "ButtonGroupsExpanded": [
      {
        "href": "/buttongroup/793",
        "Parent": {
          "href": "/device/781"
        },
        "SortOrder": 0,
        "ProgrammingType": "Freeform",
        "Buttons": [
          {
            "href": "/button/794",
            "ButtonNumber": 1,
            "ProgrammingModel": {
              "href": "/programmingmodel/795",
              "ProgrammingModelType": "AdvancedToggleProgrammingModel"
            },
            "Parent": {
              "href": "/buttongroup/793"
            },
            "Name": "Button 1",
            "Engraving": {
              "Text": "All on"
  },
  "AssociatedLED": {
    "href": "/led/788"
  }
},
{
  "href": "/button/798",
  "ButtonNumber": 2,
  "ProgrammingModel": {
    "href": "/programmingmodel/1847",
    "ProgrammingModelType": "SingleActionProgrammingModel"
  },
  "Parent": {
    "href": "/buttongroup/793"
  },
  "Name": "Button 2",
  "Engraving": {
    "Text": "Dim lock"
  },
  "AssociatedLED": {
    "href": "/led/789"
  }
},
{
  "href": "/button/802",
  "ButtonNumber": 3,
  "ProgrammingModel": {
    "href": "/programmingmodel/803",
    "ProgrammingModelType": "AdvancedToggleProgrammingModel"
  },
  "Parent": {
    "href": "/buttongroup/793"
  },
  "Name": "Button 3",
  "Engraving": {
    "Text": "Dim"
  },
  "AssociatedLED": {
    "href": "/led/790"
  }
},
{
  "href": "/button/806",
  "ButtonNumber": 4,
  "ProgrammingModel": {
    "href": "/programmingmodel/807",
    "ProgrammingModelType": "AdvancedToggleProgrammingModel"
  },
  "Parent": {
    "href": "/buttongroup/793"
  },
  "Name": "Button 4",
  "Engraving": {
    "Text": ""
  },
  "AssociatedLED": {
    "href": "/led/791"
```

```
            }
          },
          {
            "href": "/button/810",
            "ButtonNumber": 5,
            "ProgrammingModel": {
              "href": "/programmingmodel/811",
              "ProgrammingModelType": "AdvancedToggleProgrammingModel"
            },
            "Parent": {
              "href": "/buttongroup/793"
            },
            "Name": "Button 5",
            "Engraving": {
              "Text": ""
            },
            "AssociatedLED": {
              "href": "/led/792"
            }
          },
          {
            "href": "/button/814",
            "ButtonNumber": 18,
            "Parent": {
              "href": "/buttongroup/793"
            },
            "Name": "Button 18"
          },
          {
            "href": "/button/816",
            "ButtonNumber": 19,
            "Parent": {
              "href": "/buttongroup/793"
            },
            "Name": "Button 19"
          }
        ]
      }
    ]
  }
}
```

Button Action Command Processor

Caséta	✓
RA2Select	✓
HomeWorks	✓
RadioRA 3	✓

Integrator Type	Supported

Cert-Based	✓
Username/Password	✓

Supported Actions	What does it do?
PressAndHold	Macro that combines the Press and Hold action. Sending this macro will either activate the Press or the Hold programming depending on the programming on the button. Note: For buttons with Hold programming, the Hold preset is triggered automatically. However, in the future it will be updated to trigger the Hold programming after Hold time has elapsed.
PressAndRelease	Macro that combines the Press and Release actions. Sending this macro activates the Press and Release programming.
Release	Activates the Release programming.
MultiTap	Activates the DoubleTap programming.

Client triggers PressAndHold action on Button:

```
{
  "CommuniqueType": "CreateRequest",
  "Header": {
    "Url": "/button/728/commandprocessor"
  },
  "Body": {
    "Command": {
      "CommandType": "PressAndHold"
    }
  }
}
{
  "CommuniqueType": "CreateResponse",
  "Header": {
    "StatusCode": "204 NoContent",
    "Url": "button/728/commandprocessor"
  }
}
```

Master Device sends Create Response:

```
{  "CommuniqueType": "CreateResponse",
  "Header": {
    "StatusCode": "204 NoContent",
    "Url": "button/728/commandprocessor"
  }
}
```

Client triggers PressAndRelease action on Button:

```
{  "CommuniqueType": "CreateRequest",
  "Header": {
    "Url": "/button/728/commandprocessor"
  },
  "Body": {
    "Command": {
      "CommandType": "PressAndRelease"
    }
  }
}
```

Master Device sends Create Response:

```
{
  "CommuniqueType": "CreateResponse",
  "Header": {
    "StatusCode": "204 NoContent",
    "Url": "button/728/commandprocessor"
  }
}
```

Client triggers Release action on Button:

```
{
  "CommuniqueType": "CreateRequest",
  "Header": {
    "Url": "/button/728/commandprocessor"
  },
  "Body": {
    "Command": {
      "CommandType": "Release"
    }
  }
}
```

Master Device sends Create Response:

```
{
  "CommuniqueType": "CreateResponse",
  "Header": {
    "StatusCode": "204 NoContent",
    "Url": "button/728/commandprocessor"
  }
}
```

Client triggers MultiTap action on Button:

```
{
  "CommuniqueType": "CreateRequest",
  "Header": {
    "Url": "/button/728/commandprocessor"
  },
  "Body": {
    "Command": {
      "CommandType": "MultiTap"
    }
  }
}
```

Master Device sends Create Response:

```
{
  "CommuniqueType": "CreateResponse",
  "Header": {
    "StatusCode": "204 NoContent",
    "Url": "button/728/commandprocessor"
  }
}
```

Button Status Event Resource

Property	Required	Type	Details
Button	✓	HyperReference	A href to a button object.
ButtonEvent		ButtonEvent	Describes the event type of button.

ButtonEvent

Property	Required	Type	Details
EventType	✓	EventType	This value determines which other properties may or may not appear in this resource.

EventType Enum

Value
Press

Release
MultiTap
LongHold

Supported events

Programming on Button	Possible Notifications			
	Press	Release	MultiTap	LongHold
Single Action/Toggle/Advanced Toggle				
1. Press On	✓			
1. Press On 2. DoubleTap	✓		✓	
1. Press On 2. Cycle-Dim **or** Hold		✓		✓
1. Press On 2. DoubleTap 3. Cycle-Dim **or** Hold		✓	✓	✓
Dual Action				
1. Press 2. Release	✓	✓		
1. Press 2. Release 3. DoubleTap	✓	✓	✓	
Conditional				
1. Press On 2. DoubleTap 3. Cycle-Dim **or** Hold	✓	✓	✓	✓
Advanced Conditional				

1. Press On 2. DoubleTap 3. Cycle-Dim **or** Hold	✓	✓	✓	✓
Single Scene/Master Raise/Lower				
1. Raise/Lower	✓	✓		

Button Status Event Subscribe

Subscribing to individual button events

Caséta	✓
RA2Select	✓
HomeWorks	✓
RadioRA 3	✓

Integrator Type	Supported
Cert-Based	✓
Username/Password	✓

Client sends a Button Subscribe Request:

```
{
  "CommuniqueType": "SubscribeRequest",
  "Header": {
    "Url": "/button/146"
  }
}
```

Master Device sends Subscribe Response:

```
{
  "CommuniqueType": "SubscribeResponse",
  "Header": {
    "StatusCode": "204 NoContent",
    "Url": "/button/146"
  }
}
```

Master Device sends asynchronous response:

```
{
  "CommuniqueType": "UpdateResponse",
  "Header": {
    "MessageBodyType": "OneAreaDefinition",
    "StatusCode": "200 OK",
    "Url": "/button/146"
  },
  "Body": {
    "ButtonStatus": {
      "Button": {
        "href": "/button/146"
      },
      "ButtonEvent": {
        "EventType": "Press"
      }
    }
  }
}
```

Subscribing to button events for a Device

Caséta	✓
RA2Select	✓
HomeWorks	✓
RadioRA 3	✓

Integrator Type	**Supported**
Cert-Based	✓
Username/Password	✓

Client sends a single Button Status Event Subscribe Request:

```
{
  "CommuniqueType": "SubscribeRequest",
  "Header": {
    "Url": "/device/1234/button/status/event"
  }
}
```

Master Device sends Subscribe Response:

```
{
  "CommuniqueType": "SubscribeResponse",
  "Header": {
    "StatusCode": "204 NoContent",
    "Url": "/device/1234/button/status/event"
  }
}
```

Master Device sends asynchronous response:

```
{
  "CommuniqueType": "UpdateResponse",
  "Header": {
    "MessageBodyType": "MultipleButtonStatusEvent",
    "StatusCode": "200 OK",
    "Url": "/device/1234/button/status/event"
  },
  "Body": {
    "ButtonStatuses": [
      {
        "Button": {
          "href": "/button/146"
        },
        "ButtonEvent": {
          "EventType": "Press"
        }
      }
    ]
  }
}
```

LEDs

Caséta	✗	
RA2 Select	✗	
HomeWorks	✓	List read on /LED is not supported. Refer this to discover LEDs.
RadioRA 3	✓	List read on /LED is not supported. Refer this to discover LEDs.

Integrator Type	**Supported**
Cert-Based	✓
Username/Password	✓

LED Resource

Property	**Required**	**Type**	**Details**
href	✓	string	A unique identifier for this resource
Parent		HyperReference	A href to the parent control station. If this field is empty then the resource has no parent.

LED Reads

```
{
  "CommuniqueType": "ReadRequest",
  "Header": {
    "Url": "/led/789"
  }
}
```

LED Reads

```
{
  "CommuniqueType": "ReadResponse",
  "Header": {
    "MessageBodyType": "OneLEDDefinition",
    "StatusCode": "200 OK",
    "Url": "/led/789"
  },
  "Body": {
    "LED": {
      "href": "/led/789",
      "Parent": {
        "href": "/device/781"
      }
    }
  }
}
```

LED Status Reads

LED Status Resource

Property	Required	Type	Details
href	✓	string	A unique identifier for this resource
LED	✓	HyperReference	A href to the LED.
State		string	Describes the current state of the LED. Values can be 1. **On** 2. **Off** 3. **NormalFlash** 4. **RapidFlash** 5. **Unknown** : This could be received on a Read but never as an async notification

```
{
  "CommuniqueType": "ReadRequest",
  "Header": {
    "Url": "/led/789/status"
  }
}
```

```
{
  "CommuniqueType": "ReadResponse",
  "Header": {
    "MessageBodyType": "OneLEDStatus",
    "StatusCode": "200 OK",
    "Url": "/led/789"
  },
  "Body": {
    "LEDStatus": {
      "href": "/led/789/status",
      "LED": {
        "href": "/led/789"
      },
      "State": "Off"
    }
  }
}
```

Update LED State

```
{
  "CommuniqueType": "UpdateRequest",
  "Header": {
    "Url": "/led/788/status"
  },
  "Body": {
    "LEDStatus": {
      "State": "On"
    }
  }
}
{
  "CommuniqueType": "UpdateResponse",
  "Header": {
    "MessageBodyType": "OneLEDStatus",
    "StatusCode": "200 OK",
    "Url": "/led/788/status"
  },
  "Body": {
    "LEDStatus": {
      "href": "/led/788/status",
      "LED": {
        "href": "/led/788"
      },
      "State": "On"
    }
  }
}
```

LED Subscribe

```
{
  "CommuniqueType": "SubscribeRequest",
  "Header": {
    "Url": "/led/788/status"
  }
}
{
  "CommuniqueType": "SubscribeResponse",
  "Header": {
    "MessageBodyType": "OneLEDStatus",
    "StatusCode": "200 OK",
    "Url": "/led/788/status"
  },
  "Body": {
    "LEDStatus": {
      "href": "/led/788/status",
      "LED": {
        "href": "/led/788"
      },
      "State": "On"
    }
  }
}
```

ProgrammingModels

System Availability

Caséta	✓	
RA2 Select	✓	
Homeworks	✓	⚠Read on /programmingmodel is not supported.
RadioRA 3	✓	⚠Read on /programmingmodel is not supported.

Integrator Type	**Supported**	
Cert-Based	✓	
Username/Password	✓	⚠Read on /programmingmodel is not supported.

Programming models describe the the structure of the programming for buttons, virtual buttons, and timeclock events. For example, if a button has a SingleActionProgrammingModel then the same programming gets activated on every tap. In contrast, if it has a

DualActionProgrammingModel then certain programming gets activated on the press and different programming gets activated on the release. ProgrammingModelType can be used to determine what actions will be available for pressing buttons.

Programming Model Resource

Property	Required	Type	Details
href	✓	string	A unique identifier for this resource
Parent		HyperReference	The parent of this programming model. It can be a virtual button, button group, button, or timeclock event. When the parent is deleted, this programming model is deleted.
Direction		string	Direction ("Raise" or "Lower") of a raise/lower programming model. Only appears if ProgrammingModelType is "SingleSceneRaiseProgrammingModel", "SingleSceneLowerProgrammingModel", "MasterRaiseProgrammingModel", or "MasterLowerProgrammingModel"
ProgrammingModelType		ProgrammingModelType Enum	Describes the style of programming rules. This value determines which other properties may or may not appear in this resource.
Preset		HyperReference	A href to the preset object.
DualActionProperties		DualActionProperties	Describes the properties if programming model type is DualActionProgrammingModel.

AdvancedToggleProperties		AdvancedToggleProperties	Describes the properties if programming model type is AdvancedToggleProgrammingModel.

DualActionProperties

Property	Required	Type	Details
PressPreset		HyperReference	A href to the press preset.
ReleasePreset		HyperReference	

Value	**Description**
AdvancedToggleProgrammingModel	Programming for a button where different programming is activated for an "on" and "off" state.
SingleSceneRaiseProgrammingModel, SingleSceneLowerProgrammingModel	Programming for a button where a set of zones will move up or down when pressed and will stop when released.
MasterRaiseProgrammingModel, MasterLowerProgrammingModel	⚠ **Caséta, RA2Select doesn't support it. HomeWorks and RadioRA 3 don't report it.** Programming for a button where the last adjusted zone will move up or down when pressed and will stop when released.
DualActionProgrammingModel	Programming for a button that activates certain programming on a button press and different programming on a button release.
SimpleConditionalProgrammingModel	⚠ **Caséta, RA2Select doesn't support it. HomeWorks and RadioRA 3 don't report it.** Programming for a button that allows the button to change programming based on a single state variable.

Value	Description
AdvancedConditionalProgrammingModel	⚠ **Caséta, RA2Select doesn't sup port it. Homeorks and RadioRA 3 don't report it.** Programming for a button that allows complex conditional programming for different button presses.

Programming Model Reads

Client sends all ProgrammingModel Definition Read Request:

```
{
  "CommuniqueType":"ReadRequest",
  "Header":{
    "Url":"/programmingmodel"
  }
}
```

Master Device sends Read Response:

```
{  "CommuniqueType": "ReadResponse",
  "Header": {
    "MessageBodyType": "MultipleProgrammingModelDefinition",
    "StatusCode": "200 OK",
    "Url": "/programmingmodel"
  },
  "Body": {
    "ProgrammingModels": [
      {
        "href": "/programmingmodel/1",
        "Parent": {
          "href": "/parent-href"
        },
        "ProgrammingModelType": "SingleActionProgrammingModel",
        "Preset": {
          "href": "/preset/1",

        },

      },
      {
        "href": "/programmingmodel/2",
        "Parent": {
          "href": "/parent-href"
        },
        "ProgrammingModelType": "DualActionProgrammingModel",
        "DualActionPropertes": {
          "PressPreset": {
            "href": "/preset/1"
          },
          "ReleasePreset": {
            "href": "/preset/2"
          },

        },

      },
      {
        "href": "/programmingmodel/3",
        "Parent": {
          "href": "/parent-href"
        },
        "ProgrammingModelType": "AdvancedToggleProgrammingModel",
        "AdvancedTogglePropertes": {
          "PrimaryPreset": {
            "href": "/preset/1"
          },
          "SecondaryPreset": {
            "href": "/preset/2"
          },

        },

      },
      {
        "href": "/programmingmodel/4",
        "Parent": {
          "href": "/parent-href"
        },
        "Direction": "Raise",
        "ProgrammingModelType": "SingleSceneRaiseProgrammingModel",
        "AdvancedTogglePropertes": {
          "PrimaryPreset": {
            "href": "/preset/1"
          },
          "SecondaryPreset": {
            "href": "/preset/2"
          },

        },

      },

    ]
  }
}
```

Client sends one ProgrammingModel Definition Read Request:

```
{   "CommuniqueType":"ReadRequest",
  "Header":{
    "Url":"/programmingmodel/1"
  }
}
```

Master Device sends Read Response:

```
{
  "CommuniqueType": "ReadResponse",
  "Header": {
    "MessageBodyType": "OneProgrammingModelDefinition",
    "StatusCode": "200 OK",
    "Url": "/programmingmodel/1"
  },
  "Body": {
    "ProgrammingModel": {
      "href": "/programmingmodel/1",
      "Parent": {
        "href": "/parent-href"
      },
      "ProgrammingModelType": "SingleActionProgrammingModel",
      "Preset": {
        "href": "/preset/1"
      },

    }
  }
}
```

Presets

System Availability

Caséta	✓	List read on /preset is not supported.
RA2 Select	✓	List read on /preset is not supported.
Homeworks	✓	List read on /preset is not supported.
RadioRA 3	✓	List read on /preset is not supported.

Integrator Type	**Supported**
Cert-Based	✓
Username/Password	✓

Preset Resource

Due to the scale of HomeWorks and RadioRA 3 systems, none of the optional fields in the below table will show up when the client does a read to fetch information about the Preset.
Filtered Reads(Residential Systems LEAP Protocol - Control and Monitoring#ControlandMonitoring-FilterReadsonPreset) must be used to get more information about the PresetAssignments in a Preset.

Property	Required	Type	Details	Supported Systems
href	✓	string	The unique identifier for this resource.	**Caséta, RA2 Select, HomeWorks, RadioRA 3**
Parent	✓	HyperReference	The object owning this resource. This field will be omitted if no Parent exists.	**Caséta, RA2 Select, HomeWorks, RadioRA 3**
PresetAssignments		List<HyperReference>	List of preset assignments associated to this preset.	**Caséta, RA2 Select**
DimmedLevelAssignments		List<HyperReference>	List of preset assignments associated to this preset.	**Caséta, RA2 Select, HomeWorks, RadioRA 3**

ShadeLevelAssignments		List<HyperReference>	List of shade level assignments associated to this preset.	**Caséta, RA2 Select, HomeWorks, RadioRA 3**
ShadeLevelWithTiltAssignments		List<HyperReference>	List of shade level with tilt assignments associated to this preset.	**HomeWorks, RadioRA 3**
ShadeLevelWithTiltWhenClosedAssignments		List<HyperReference>	List of shade level with tilt when closed assignments associated to this preset.	**HomeWorks, RadioRA 3**
SwitchedLevelAssignments		List<HyperReference>	List of switched level assignments associated to this preset.	**Caséta, RA2 Select, HomeWorks, RadioRA 3**
FanSpeedAssignments		List<HyperReference>	List of fan speed assignments associated to this preset.	**HomeWorks, RadioRA 3**

CCOLevelAssignments		List<HyperReference>	List of CCO level assignments associated to this preset.	**HomeWorks, RadioRA 3**
ReceptacleLevelAssignments		List<HyperReference>	List of receptacle level assignments associated to this preset.	**HomeWorks, RadioRA 3**
SpectrumTuningLevelAssignments		List<HyperReference>	List of spectrum tuning assignments associated to this preset.	**HomeWorks, RadioRA 3**
WarmDimAssignments		List<HyperReference>	List of warm dim assignments associated to this preset.	**HomeWorks, RadioRA 3**
WhiteTuningLevelAssignments		List<HyperReference>	List of white tuning assignments associated to this preset.	**HomeWorks, RadioRA 3**

TiltAssignments		List<HyperReference>	List of tilt assignments associated to this preset.	**Caséta, RA2 Select**
OccupancySettingsAssignments		List<HyperReference>	List of occupancy settings assignments associated to this preset.	**Caséta, RA2 Select**
OccupancySensorSettingsAssignments		List<HyperReference>	List of occupancy sensor settings assignments associated to this preset.	**Caséta, RA2 Select**
StartSequenceAssignments		List<HyperReference>	List of start sequence assignments associated to this preset.	**Caséta, RA2 Select**
SonosPlayAssignments		List<HyperReference>	List of sonos play assignments associated to this preset.	**Caséta, RA2 Select**

FavoriteCycleAssignments		List<HyperReference>	List of favorite cycle assignments associated to this preset.	**Caséta, RA2 Select**
NextTrackAssignments		List<HyperReference>	List of next track assignments associated to this preset.	**Caséta, RA2 Select**
RaiseLowerAssignments		List<HyperReference>	List of raise lower assignments associated to this preset.	**Caséta, RA2 Select**
PauseAssignments		List<HyperReference>	List of pause assignments associated to this preset.	**Caséta, RA2 Select**
PlayPauseAssignments		List<HyperReference>	List of play pause assignments associated to this preset.	**Caséta, RA2 Select**
AssignableResources		List<HyperReference>	List of assignable resources associated to this preset.	**Caséta, RA2 Select, HomeWorks, RadioRA 3**

Preset Reads

Caséta/RA2 Select

Client sends one Preset Definition Read Request:

```
{
  "CommuniqueType":"ReadRequest",
  "Header":{
    "Url":"/preset/1"
  }
}
```

Master Device sends Read Response:

```
{
  "CommuniqueType": "ReadResponse",
  "Header": {
    "MessageBodyType": "OnePresetDefinition",
    "StatusCode": "200 OK",
    "Url": "/preset/1"
  },
  "Body": {
    "Preset": {
      "href": "/preset/1",
      "Parent": {
        "href": "/parent-href"
      },
      "PresetAssignments": [
        {
          "href": "/presetassignment/1"
        },

      ],
      "DimmedLevelAssignments": [
        {
          "href": "/presetassignment/2"
        },

      ],
      "ShadeLevelAssignments": [
        {
          "href": "/presetassignment/3"
        },

      ],
      "DimmedLevelAssignments": [
        {
          "href": "/presetassignment/2"
        },
      ],
      "SwitchedLevelAssignments": [
        {
          "href": "/presetassignment/6"
        }
      ],
      "TiltAssignments": [
        {
          "href": "/presetassignment/11"
        }
      ],
      "OccupancySettingsAssignments": [
        {
          "href": "/presetassignment/12"
        }
      ],
      "OccupancySensorSettingsAssignments": [
        {
          "href": "/presetassignment/13"
        }
      ],
      "StartSequenceAssignments": [
        {
          "href": "/presetassignment/14"
        }
      ],
      "SonosPlayAssignments": [
        {
          "href": "/presetassignment/15"
        }
      ],
      "FavoriteCycleAssignments": [
        {
          "href": "/presetassignment/16"
        }
      ],
      "NextTrackAssignments": [
        {
          "href": "/presetassignment/17"
        }
      ],
      "RaiseLowerAssignments": [
        {
          "href": "/presetassignment/18"
        }
      ],
      "PauseAssignments": [
        {
          "href": "/presetassignment/19"
        }
      ],
      "PlayPauseToggleAssignments": [
        {
          "href": "/presetassignment/20"
        }
      ],
      "AssignableResources": [
        {
          "href": "/zone/1"
        },
        {
          "href": "/device/1"
        }
      ]
    }
  }
}
```

Phoenix

Client sends one Preset Definition Read Request:

```
{
  "CommuniqueType":"ReadRequest",
  "Header":{
    "Url":"/preset/1"
  }
}
```

Master Device sends Read Response:

```
{
  "CommuniqueType": "ReadResponse",
  "Header": {
    "MessageBodyType": "OnePresetDefinition",
    "StatusCode": "200 OK",
    "Url": "/preset/1"
  },
  "Body": {
    "Preset": {
      "href": "/preset/1",
      "Parent": {
        "href": "/parent-href"
      },
        },
}
```

Filter Reads on Preset

Integrator Type	Supported
Cert-Based	✓
Username/Password	✓

DimmedLevelAssignment Resource

Property	Required	Type	Details
href	✓	string	A unique identifier for this resource.
Parent		HyperReference	A href to parent preset.
AssignableResource		HyperReference	A href to assignable resource.
FadeTime		Timespan	Specifies the amount of fade time from the current level.
DelayTime		Timespan	Specifies a delay prior to starting the fade.
Level		float	Describes the Intensity or Level.

Preset DimmedLevelAssignment Read

```
{
  "CommuniqueType": "CreateRequest",
  "Header": {
    "Url": "/preset/1/dimmedlevelassignment/commandprocessor"
  },
  "Body": {
    "Command": {
      "CommandType": "Filter",
      "FilterParameters": {
        "Where": {
          "Binary": {
            "Operator": "=",
            "Left": {
              "Resource": {
                "href": "$1/assignableresource/associatedarea"
              }
            },
            "Right": {
              "Resource": {
                "href": "/area/71"
              }
            }
          }
        }
      }
    }
  }
}
```

Master Device sends Multiple DimmedLevelAssignment Definition Response:

```
{
  "CommuniqueType": "CreateResponse",
  "Header": {
    "MessageBodyType": "MultipleDimmedLevelAssignmentDefinition",
    "StatusCode": "201 Created",
    "Url": "/preset/1/dimmedlevelassignment/commandprocessor"
  },
  "Body": {
    "DimmedLevelAssignments": [
      {
        "href": "/dimmedlevelassignment/10",
        "Parent": {
          "href": "/preset/1"
        },
        "AssignableResource": {
          "href": "/zone/2"
        },
               "Level": 5,
        "FadeTime": "2"
      },
      {
        "href": "/dimmedlevelassignment/11",
        "Parent": {
          "href": "/preset/1"
        },
        "AssignableResource": {
          "href": "/zone/3"
        },
               "Level": 20,
        "FadeTime": "2"
      },

    ]
  }
}
```

FanSpeedAssignment Resource

Property	Required	Type	Details
href	✓	string	A unique identifier for this resource.
Parent		HyperReference	A href to parent preset.
AssignableResource		HyperReference	A href to assignable resource.
DelayTime		Timespan	Specifies a delay prior to starting the fade.
Speed		FanSpeed	Describes the speed of fan. This value determines which other properties may or may not appear in this resource.

FanSpeed Enum

Value
High
MediumHigh
Medium
Low
Off

Preset FanSpeedAssignment Read

```
{
  "CommuniqueType": "CreateRequest",
  "Header": {
    "Url": "/preset/1/fanspeedassignment/commandprocessor"
  },
  "Body": {
    "Command": {
      "CommandType": "Filter",
      "FilterParameters": {
        "Where": {
          "Binary": {
            "Operator": "=",
            "Left": {
              "Resource": {
                "href": "$1/assignableresource/associatedarea"
              }
            },
            "Right": {
              "Resource": {
                "href": "/area/71"
              }
            }
          }
        }
      }
    }
  }
}
```

Master Device sends Multiple FanSpeedAssignment Definition Response:

```
{
  "CommuniqueType": "CreateResponse",
  "Header": {
    "MessageBodyType": "MultipleFanSpeedAssignmentDefinition",
    "StatusCode": "201 Created",
    "Url": "/preset/1/fanspeedassignment/commandprocessor"
  },
  "Body": {
    "FanSpeedAssignments": [
      {
        "href": "/fanspeedassignment/10",
        "Parent": {
          "href": "/preset/1"
        },
        "AssignableResource": {
          "href": "/zone/2"
        },
        "DelayTime": "2",
        "Speed": "High"
      },
      {
        "href": "/fanspeedassignment/11",
        "Parent": {
          "href": "/preset/1"
        },
        "AssignableResource": {
          "href": "/zone/3"
        },
        "DelayTime": "2",
        "Speed": "Medium"
      },

    ]
  }
}
```

ReceptacleLevelAssignment Resource

Property	Required	Type	Details
href	✓	string	A unique identifier for this resource.
Parent		HyperReference	A href to parent preset.
AssignableResource		HyperReference	A href to assignable resource.
DelayTime		Timespan	Specifies a delay prior to starting the fade.
ReceptacleLevel		string	Describes the level of receptacle. Value can be "**On**" or "**Off**".

Preset ReceptacleLevelAssignment Read

```
{

  "CommuniqueType": "CreateRequest",
  "Header": {
    "Url": "/preset/1/receptaclelevelassignment/commandprocessor"
  },
  "Body": {
    "Command": {
      "CommandType": "Filter",
      "FilterParameters": {
        "Where": {
          "Binary": {
            "Operator": "=",
            "Left": {
              "Resource": {
                "href": "$1/assignableresource/associatedarea"
              }
            },
            "Right": {
              "Resource": {
                "href": "/area/71"
              }
            }
          }
        }
      }
    }
  }
}
```

Master Device sends Multiple ReceptacleLevelAssignment Definition Response:

```
{
  "CommuniqueType": "CreateResponse",
  "Header": {
    "MessageBodyType": "MultipleReceptacleLevelAssignmentDefinition",
    "StatusCode": "201 Created",
    "Url": "/preset/1/receptaclelevelassignment/commandprocessor"
  },
  "Body": {
    "FanSpeedAssignments": [
      {
        "href": "/receptaclelevelassignment/10",
        "Parent": {
          "href": "/preset/1"
        },
        "AssignableResource": {
          "href": "/zone/2"
        },
        "DelayTime": "2",
        "ReceptacleLevel": "On"
      },
      {
        "href": "/receptaclelevelassignment/11",
        "Parent": {
          "href": "/preset/1"
        },
        "AssignableResource": {
          "href": "/zone/3"
        },
        "DelayTime": "2",
        "ReceptacleLevel": "Off"
      },

    ]
  }
}
```

SwitchedLevelAssignment Resource

Property	Required	Type	Details
href	✓	string	A unique identifier for this resource.
Parent		HyperReference	A href to parent preset.
AssignableResource		HyperReference	A href to assignable resource.
DelayTime		Timespan	Specifies a delay prior to starting the fade.
SwitchedLevel		string	Describes the level of switch. Value can be "**On**", "**Off**" and "**Unaffected**"

Preset SwitchedLevelAssignment Read

```
{
  "CommuniqueType": "CreateRequest",
  "Header": {
    "Url": "/preset/1/switchedlevelassignment/commandprocessor"
  },
  "Body": {
    "Command": {
      "CommandType": "Filter",
      "FilterParameters": {
        "Where": {
          "Binary": {
            "Operator": "=",
            "Left": {
              "Resource": {
                "href": "$1/assignableresource/associatedarea"
              }
            },
            "Right": {
              "Resource": {
                "href": "/area/71"
              }
            }
          }
        }
      }
    }
  }
}
```

Master Device sends Multiple SwitchedLevelAssignment Definition Response:

```
{
  "CommuniqueType": "CreateResponse",
  "Header": {
    "MessageBodyType": "MultipleSwitchedLevelAssignmentDefinition",
    "StatusCode": "201 Created",
    "Url": "/preset/1/switchedlevelassignment/commandprocessor"
  },
  "Body": {
    "FanSpeedAssignments": [
      {
        "href": "/switchedlevelassignment/10",
        "Parent": {
          "href": "/preset/1"
        },
        "AssignableResource": {
          "href": "/zone/2"
        },
        "DelayTime": "2",
        "SwitchedLevel": "On"
      },
      {
        "href": "/switchedlevelassignment/11",
        "Parent": {
          "href": "/preset/1"
        },
        "AssignableResource": {
          "href": "/zone/3"
        },
        "DelayTime": "2",
        "SwitchedLevel": "Off"
      },

    ]
  }
}
```

WarmDimAssignment Resource

Property	Nullable	Required	Type	Details
href	✗	✓	string	A unique identifier for this resource.
Parent	✗	✓	HyperReference	A href to parent preset.
AssignableResource	✗	✓	HyperReference	assignable
Level	✗	✗	float	Describes the Intensity or Level.
FadeTime	✗	✗	Timespan	Specifies the amount of fade time from the current level.
DelayTime	✗	✗	Timespan	Specifies a delay prior to starting the fade.
CurveDimming	✗	✓	CurveDimming	The href of a curve that a CurveDimming object follows.

Preset WarmDimAssignment Read

```
{
  "CommuniqueType": "CreateRequest",
  "Header": {
    "Url": "/preset/1/warmdimassignment/commandprocessor"
  },
  "Body": {
    "Command": {
      "CommandType": "Filter",
      "FilterParameters": {
        "Where": {
          "Binary": {
            "Operator": "=",
            "Left": {
              "Resource": {
                "href": "$1/assignableresource/associatedarea"
              }
            },
            "Right": {
              "Resource": {
                "href": "/area/71"
              }
            }
          }
        }
      }
    }
  }
}
```

Master Device sends Multiple SwitchedLevelAssignment Definition Response:

```
{
   "CommuniqueType": "CreateResponse",
   "Header": {
      "MessageBodyType": "MultipleWarmDimAssignmentDefinition",
      "StatusCode": "201 Created",
      "Url": "/preset/1/warmdimassignment/commandprocessor"
   },
   "Body": {
      "WarmDimAssignments": [
         {
            "AssignableResource": {
               "href": "/zone/924"
            },
            "Parent": {
               "href": "/preset/1"
            },
            "DelayTime": "0",
            "FadeTime": "2",
            "CurveDimming": {
               "Curve": {
                  "href": "/curve/4"
               }
            }
         },
         {
            "href": "/warmdimassignment/912",
            "Parent": {
               "href": "/preset/1"
            },
            "AssignableResource": {
               "href": "/zone/587"
            },
            "Level": 100,
            "DelayTime": "0",
            "FadeTime": "2"
         },
         {
            "href": "/warmdimassignment/914",
            "Parent": {
               "href": "/preset/1"
            },
            "AssignableResource": {
               "href": "/zone/599"
            },
            "Level": 100,
            "DelayTime": "0",
            "FadeTime": "2",
            "CurveDimming": {
               "Curve": {
                  "href": "/curve/4"
               }
            }
         },
         {
            "href": "/warmdimassignment/916",
            "Parent": {
               "href": "/preset/1"
            },
            "AssignableResource": {
               "href": "/zone/610"
            },
            "Level": 100,
            "DelayTime": "0",
            "FadeTime": "2",
            "CurveDimming": {
               "Curve": {
                  "href": "/curve/4"
               }
            }
         }
      ]
   }
}
```

SpectrumTuningLevelAssignment Resource

Property	Required	Type	Details
href	✓	string	A unique identifier for this resource.
Parent		HyperReference	A href to parent preset.
AssignableResource		HyperReference	A href to assignable resource.
Level		float	Describes the Intensity or Level.

NaturalShowAffectsIntensity		bool	true if there is a natural show programmed to the zone that controls the level/intenstiy
DelayTime		Timespan	Specifies a delay prior to starting the fade.
FadeTime		Timespan	Specifies the amount of fade time from the current level.
Vibrancy		float	Specifies the amount of white light that composes the color output.
ColorTuningStatus		ColorTuningStatus	Specifies the status of color. This value determines which other properties may or may not appear in this resource.

ColorTuningStatus

Only one of the following properties must be set in a GoToDimmedLevel or GoToSpectrumTuning command. Unlike other LEAP commands omitting one of the properties below in a request necessarily leave the others unchanged.

Property	Type	Details

HSVTuningLevel	HSVTuningLevel	Describes the desired color using Hue and Saturation. Hue must be in the range of 0-359, while Saturation must be between 0 and 100. Both will be rounded to the nearest whole number (no decimals). Can only be present when AvailableControlTypes contains ColorTune. If omitted from the command this will be omitted from command response
WhiteTuningLevel	WhiteTuningLevel	Describes the desired color using the Kelvin scale, in increments of 50 Kelvin. For zones that support vibrancy, vibrancy will have the most impact when this property is used for color control. Can only be present when AvailableControlTypes contains WhiteTune. **A zone with an AvailableControlType of WhiteTune may not necessarily have a corrolated color temparature (CCT) at all times if it supports other types of color control such as ColorTune. This property will be omitted in responses if the zone is not currently on the black body curve.** If omitted from the command this will be omitted from command response. Minimum Value: 1400K Maximum Value: 10000K
XYTuningLevel	XYTuningLevel	Describes the desired color using x and y axis from the CIE 1931 color space. X and Y are float values between 0 and 1. Can only be present when AvailableControlTypes contains ColorTune. If omitted from the command this will be omitted from command response.

CurveDimming	CurveDimming	Representation of any type of curve dimming, such as a warm dim curve. Can only be present when AvailableControlTypes contains WarmDim. This property will be omitted in responses if the zone is not (or is no longer) following a dimming curve. If the zone is explicitly commanded to a color or color temperature (either through a LEAP command or preset activation), it will automatically stop following its dimming curve.
NaturalShow	NaturalShow	Show that controls the CCT and optionally the intensity of a Ketra load through the day. This is currently a read-only property.

The limits specified in the Level specific tables are enforced by the protocol. However, for more information about the recommended limits so you can provide the best experience with Ketra lamps, refer the following documents:

1. CIE1931-ColorChart_042319.ai
2. CIE1931-ColorChart-Layers_070919.ai

XYTuningLevel

Property	Required	Type	Details	Protocol Limits
X	✓	float	Value of X	must be the range 0 to 1
Y	✓	float	Value of Y	must be the range 0 to 1

HSVTuningLevel

Property	Required	Type	Details
Hue	✓	float	Hue is a degree on the color wheel from 0 to 359.
Saturation	✓	float	Saturation is a percentage value; 0% means a shade of gray and 100% is the full color.

WhiteTuningLevel

Property	Required	Type	Details

```
        }
      }
    }
  }
}
```

Master Device sends Multiple SpectrumTuningLevelAssignment Definition Response:

```
{
  "CommuniqueType": "CreateResponse",
  "Header": {
    "MessageBodyType": "MultipleSpectrumTuningLevelAssignmentDefinition",
    "StatusCode": "201 Created",
    "Url": "/preset/746/spectrumtuninglevelassignment/commandprocessor"
  },
  "Body": {
    "SpectrumTuningLevelAssignments": [
      {
        "href": "/spectrumtuninglevelassignment/1429",
        "Parent": {
          "href": "/preset/746"
        },
        "AssignableResource": {
          "href": "/zone/1366"
        },
        "Level": 100,
        "DelayTime": "0",
        "FadeTime": "2",
        "Vibrancy": 25,
        "ColorTuningStatus": {
          "XYTuningLevel": {
            "X": 0.15,
            "Y": 0.6925
          },
          "HSVTuningLevel": {
            "Hue": 140,
            "Saturation": 89
          }
        }
      },
      {
        "href": "/spectrumtuninglevelassignment/1431",
        "Parent": {
          "href": "/preset/746"
        },
        "AssignableResource": {
          "href": "/zone/1383"
        },
        "Level": 0,
        "DelayTime": "0",
        "FadeTime": "2"
      },
      {
        "href": "/spectrumtuninglevelassignment/1433",
        "Parent": {
          "href": "/preset/746"
        },
        "AssignableResource": {
          "href": "/zone/1423"
        },
        "Level": 75,
        "DelayTime": "0",
        "FadeTime": "2",
        "Vibrancy": 33
      }
    ]
  }
}
```

Preset SpectrumTuningLevelAssignment Read v2 - includes preset assignments that activate a natural show

```
{
  "CommuniqueType": "CreateRequest",
  "Header": {
    "Url": "/preset/1/v2/spectrumtuninglevelassignment/commandprocessor"
  },
  "Body": {
    "Command": {
      "CommandType": "Filter",
      "FilterParameters": {
        "Where": {
          "Binary": {
            "Operator": "=",
            "Left": {
              "Resource": {
                "href": "$1/assignableresource/associatedarea"
              }
            },
            "Right": {
              "Resource": {
                "href": "/area/71"
              }
            }
          }
        }
      }
    }
  }
}
```

Master Device sends Multiple SpectrumTuningLevelAssignment Definition Response:

```
{
  "CommuniqueType": "CreateResponse",
  "Header": {
    "MessageBodyType": "MultipleSpectrumTuningLevelAssignmentDefinition",
    "StatusCode": "201 Created",
    "Url": "/preset/746/spectrumtuninglevelassignment/commandprocessor"
  },
  "Body": {
    "SpectrumTuningLevelAssignments": [
      {
        "href": "/spectrumtuninglevelassignment/1429",
        "Parent": {
          "href": "/preset/746"
        },
        "AssignableResource": {
          "href": "/zone/1366"
        },
        "Level": 100,
        "DelayTime": "0",
        "FadeTime": "2",
        "Vibrancy": 25,
        "ColorTuningStatus": {
          "NaturalShow":"/naturalshow/231"
        }
      },
          {
        "href": "/spectrumtuninglevelassignment/1429",
        "Parent": {
          "href": "/preset/746"
        },
        "AssignableResource": {
          "href": "/zone/1386"
        },
        "Level": 100,
        "DelayTime": "0",
        "FadeTime": "2",
        "Vibrancy": 25,
        "ColorTuningStatus": {
          "NaturalShow":"/naturalshow/238"
        }
      },
    ]
  }
}
```

WhiteTuningLevelAssignment Resource

Integrator Type	Supported
Cert-Based	✓
Username/Password	✓

Property	Required	Type	Details
href	✓	string	A unique identifier for this resource.

Parent			A href to parent preset.
AssignableResource		HyperReference	A href to assignable resource.
Level		float	Describes the Intensity or Level.
DelayTime		Timespan	Specifies a delay prior to starting the fade.
FadeTime		Timespan	Specifies the amount of fade time from the current level.
WhiteTuningLevel		WhiteTuningLevel	Describes the desired color using the Kelvin scale, in increments of 10K.

Preset WhiteTuningLevelAssignment Read

```
{
  "CommuniqueType": "CreateRequest",
  "Header": {
    "Url": "/preset/1/whitetuninglevelassignment/commandprocessor"
  },
  "Body": {
    "Command": {
      "CommandType": "Filter",
      "FilterParameters": {
        "Where": {
          "Binary": {
            "Operator": "=",
            "Left": {
              "Resource": {
                "href": "$1/assignableresource/associatedarea"
              }
            },
            "Right": {
              "Resource": {
                "href": "/area/71"
              }
            }
          }
        }
      }
    }
  }
}
```

Master Device sends Multiple WhiteTuningLevelAssignment Definition Response:

```
{
  "CommuniqueType": "CreateResponse",
  "Header": {
    "MessageBodyType": "MultipleWhiteTuningLevelAssignmentDefinition",
    "StatusCode": "201 Created",
    "Url": "/preset/1/whitetuninglevelassignment/commandprocessor"
  },
  "Body": {
    "WhiteTuningLevelAssignments": [
      {
        "href": "/whitetuninglevelassignment/1429",
        "Parent": {
          "href": "/preset/1"
        },
        "AssignableResource": {
          "href": "/zone/1366"
        },
        "Level": 100,
        "DelayTime": "0",
        "FadeTime": "2",
        "Vibrancy": 25,
        "WhiteTuningLevel":{
                       "Kelvin": 10000
               }
      },
      {
        "href": "/whitetuninglevelassignment/1431",
        "Parent": {
          "href": "/preset/1"
        },
        "AssignableResource": {
          "href": "/zone/1383"
        },
        "Level": 100,
        "DelayTime": "0",
        "FadeTime": "2"
      }
    ]
  }
}
```

CCOLevelAssignment Resource

Property	Required	Type	Details
href	✓	string	A unique identifier for this resource.
Parent		HyperReference	A href to parent preset.
AssignableResource		HyperReference	A href to assignable resource.
DelayTime		Timespan	Specifies a delay prior to starting the fade.
CCOLevel		string	Describes the level of CCO. Value can be "**Open**" or "**Closed**".

Preset CCOLevelAssignment Read

```
{
  "CommuniqueType": "CreateRequest",
  "Header": {
    "Url": "/preset/1/ccolevelassignment/commandprocessor"
  },
  "Body": {
    "Command": {
      "CommandType": "Filter",
      "FilterParameters": {
        "Where": {
          "Binary": {
            "Operator": "=",
            "Left": {
              "Resource": {
                "href": "$1/assignableresource/associatedarea"
              }
            },
            "Right": {
              "Resource": {
                "href": "/area/71"
              }
            }
          }
        }
      }
    }
  }
}
```

Master Device sends Multiple CCOLevelAssignment Definition Response:

```
{
  "CommuniqueType": "CreateResponse",
  "Header": {
    "MessageBodyType": "MultipleCCOLevelAssignmentDefinition",
    "StatusCode": "201 Created",
    "Url": "/preset/1/ccolevelassignment/commandprocessor"
  },
  "Body": {
    "CCOLevelAssignments": [
      {
        "href": "/ccolevelassignment/10",
        "Parent": {
          "href": "/preset/1"
        },
        "AssignableResource": {
          "href": "/zone/10"
        },
        "DelayTime": "0",
        "CCOLevel": "Open"
      }
    ]
  }
}
```

Discovering buttons and button groups

Integrator Type	Supported
Cert-Based	✓
Username/Password	✓

Read should be on device href in the AssociatedGangedDevices list.

```
{
  "CommuniqueType": "ReadRequest",
  "Header": {
    "Url": "/device/781/buttongroup"
  }
}
{
  "CommuniqueType": "ReadResponse",
  "Header": {
    "MessageBodyType": "MultipleButtonGroupDefinition",
    "StatusCode": "200 OK",
    "Url": "/device/781/buttongroup"
  },
  "Body": {
    "ButtonGroups": [
      {
        "href": "/buttongroup/793",
        "Parent": {
          "href": "/device/781"
        },
        "SortOrder": 0,
        "StopIfMoving": "Disabled",
        "ProgrammingType": "Freeform",
        "Buttons": [
          {
            "href": "/button/794"
          },
          {
            "href": "/button/798"
          },
          {
            "href": "/button/810"
          }
        ]
      }
    ]
  }
}
```

Activating a Preset

Caséta	✓
RA2 Select	✓
HomeWorks	✓
RadioRA 3	✓

Integrator Type	Supported
Cert-Based	✓
Username/Password	✗

Property	Required	Type	Details
href	✓	string	The unique identifier for this resource.

Client sends a request to activate a Preset

```
{
  "CommuniqueType": "CreateRequest",
  "Header": {
    "Url": "<preset-href>/commandprocessor",
    "ClientTag": "your-client-tag-here"
  },
  "Body": {
    "Command": {
      "CommandType": "Activate"
    }
  }
}
```

Server responds when the Preset was activated

```
{
    "CommuniqueType":"CreateResponse",
    "Header":{
        "StatusCode":"204 NoContent",
        "Url": "<preset-href>/commandprocessor"
    }
}
```

Zone

Caséta	✓	
RA2 Select	✓	
HomeWorks	⚠	List read on /zone is not supported. Read this to discover zones.
RadioRA 3	⚠	List read on /zone is not supported. Read this to discover zones.

Integrator Type	Supported	
Cert-Based	✓	⚠Read on /zone is not supported for HomeWorks and RadioRA 3.
Username/Password	✓	⚠Read on /zone is not supported.

Zone Resource

A zone represents the smallest controllable end-point over LEAP.

Property	Nullable	Required	Type	Details
href		✓	string	The unique identifier for this resource.

Name		✓	string	This is the "friendly" name of the zone.
Device			HyperReference	Typically only provided when the Device to Zone relationship is one-to-one, such as in systems like Caséta and Ra2 Select. It refers to the physical dimmer, switch, or shade this zone is attached to.
ControlType		✓	ControlType	The primary control type. Clients should use AvailableControlTypes instead of ControlType if and only if AvailableControlTypes contains 'Dimmed'.
AvailableControlTypes	✗	✓	List<AvailableControlType >	A list that provides the categories of control this zone supports. **This property should be ignored entirely if it does not contain 'Dimmed'.**
ThirdPartyIdentification			ThirdPartyIdentification	Describes the third-party-supplied identification for the zone, if applicable.

AssociatedArea			HyperReference	In Caséta and Ra2 Select this field is omitted and instead the AssociatedArea of the Device should be used to organize zones/devices.
AssociatedFacade			HyperReference	In Caséta this field will only be provided for zones that are tilt only shades.
ColorTuningProperties			Residential Systems LEAP Protocol - Control and Monitoring#ControlandMonitoring-ColorTuningProperties	This field will only be present for zones that are capable of color control. Eg. SpectrumTune
SingleSetPointHVACProperties				This field will only be present for SingleSetPointHVAC zones.
DualSetPointHVACProperties				This field will only be present for DualSetPointHVAC zones.
DefaultDimCurve	❌	❌	HyperReference	This field will be present for zones that support CurveDimming. This is the curve that a zone should send when applying CurveDimming.

ControlType Enum*

Value	Description	Supported Commands on this Zone Type

Dimmed	A dimmable zone, capable of going from 0% to 100% in 1% increments.	GoToDimmedLevel, Raise, Lower, Stop
Switched	A switchable zone, only capable of going On and Off.	GoToSwitchedLevel
SpectrumTune	A zone capable of going to a color, vibrancy, and brightness. This control type is commonly used to describe control of Ketra products.	GoToSpectrumTuningLevel, Raise, Lower, Stop
WhiteTune	A zone capable of going to brightness and color correlated temperature.	GoToWhiteTuningLevel
WarmDim	A zone capable of Warm Dimming.	GoToWarmDim
Shade	A shading zone, capable of going from 0% to 100% in 0.01% increments.	GoToShadeLevel, Raise, Lower, Stop
ShadeWithTilt	A shading zone, capable of going from 0% to 100% in lift and tilt.	GoToShadeLevelWithTilt, Raise, Lower, Stop
ShadeWithTiltWhenClosed	A shading zone, capable of going from 0% to 100% in lift when tilt is 0% and 0% to 50% in tilt when lift is 0%	GoToShadeLevelWithTiltWhenClosed, Raise, Lower, Stop
Receptacle	A receptacle zone	GoToReceptacleLevel
CCO	A CCO zone	GoToCCOLevel
FanSpeed	A fan speed zone, capable of a set of speeds.	GoToFanSpeed
Tilt	A tilt only shading zone, capable of going from 0% to 100% in tilt.	GoToTilt, Raise, Lower, Stop

OpenCloseStop	A motor zone, capable of Raise/Lower/Stop shades that are controlled by a motor.	Raise, Lower, Stop
SingleSetPointHVAC	A zone that is capable of Heating, Ventilation and Air Conditioning. The zone continually strives to maintain the temperature at a single set point based on the operating mode it is in.	
DualSetPointHVAC	A zone that is capable of Heating, Ventilation and Air Conditioning. The zone continually strives to maintain the temperature between two set points based on the operating mode it is in.	

AvailableControlType

Value	Description
Dimmed	A dimmable zone capable of going from 0% to 100% in 1% increments
ColorTune	A zone that is capable of color control.
Vibrancy	A zone that is capable of vibrancy control.
WhiteTune	A zone that is capable of going to a color correlated temperature (CCT). **The presence of this AvailableControlType does not imply that the zone in question will report a CCT at all times.**
WarmDim	A zone that is capable of warm dimming.

ThirdPartyIdentification

Property	Required	Type	Details
Id	✓	string	A third-party-supplied identifier.

WhiteTuningLevelRange

The range for WhiteTuningLevel for a resource.

Property	Required	Type	Details
Min	True	float64	Minimum WhiteTuningLevel supported for a resource.
Max	True	float64	Maximum WhiteTuningLevel supported for a resource.

ColorTuningProperties

Property	Required	Type	Details
WhiteTuningLevelRange	True	Residential Systems LEAP Protocol - Control and Monitoring#ControlandMonitoring-WhiteTuningLevelRange	The range for WhiteTuningLevel for a resource.

SingleSetPointHVACProperties

Property	Required	Type	Details
SetPointRange	✓	SetPointRange	Contains the allowed range and the step size for the single set point HVAC zone.
TemperatureScales	✓	List<TemperatureUnit>	These are the native scales of the thermostat. The server will neither report nor accept non-native scales.
OperatingModes	✓	List<OperatingMode>	
FanModes	✓	List<FanMode>	
EcoModeProperties		EcoModeProperties	If EcoModeProperties is absent in the zone definition, then it is not supported.

DualSetPointHVACProperties

Property	Required	Type	Details
HeatingSetPointRange	✓	SetPointRange	Contains the allowed range and the step size for the heating set point on the HVAC zone.

CoolingSetPointRange	✓	SetPointRange	Contains the allowed range and the step size for the heating set point on the HVAC zone.
HeatingCoolingDelta	✓	SetPointRange	The allowed delta between the max cooling and min heating set points.
TemperatureScales	✓	List<TemperatureUnit>	These are the native scales of the thermostat. The server will neither report nor accept non-native scales.
OperatingModes	✓	List<OperatingMode>	
FanModes	✓	List<FanMode>	
EcoModeProperties		EcoModeProperties	If EcoModeProperties is absent in the zone definition, then it is not supported.

SetPointRange

Property	Required	Type	Details
F		Range	The allowed range on the Fahrenheit scale
C		Range	The allowed range on the Celsius scale

Range

Property	Required	Type
Min		float
Max		float
Step		float

Temperature Unit Enum

Value
Fahrenheit
Celsius

Fan Mode Enum

Value
Auto
On
Off
Cycle
High
Medium
Low
Top

Operating Mode Enum

Value
Off
Heat
Cool
Auto
EmergencyHeat
Fan
Dry

EcoModeProperties

Property	Required	Type	Details
ActiveState	✓	ActiveState	Indicates if Eco mode is active right now.
EnabledState		EnabledState	If enabled EcoMode can be enabled, if disabled the server will not allow the client to enable EcoMode.

ActiveState enum

Value
Active
Inactive

Unknown

Zone Reads

Client sends All Zones Definition Read Request:

```
{
        "CommuniqueType": "ReadRequest",
        "Header": {
                "Url": "/zone"
        }
}
```

Bridge Send Read Response:

```
{
  "CommuniqueType": "ReadResponse",
  "Header": {
    "MessageBodyType": "MultipleZoneDefinition",
    "StatusCode": "200 OK",
    "Url": "/zone"
  },
  "Body": {
    "Zones": [
      {
        "Name": "Kitchen Light",
        "href": "/zone/1",
        "ControlType": "Dimmed",
        "Device": {
          "href": "/device/4"
        }
      },
      {
        "href": "/zone/2",
        "Name": "Hallway HVAC Zone Single",
        "ControlType": "SingleSetPointHVAC",
                "ThirdPartyIdentification" : {
          "Id": "101"
        },
        "SingleSetPointHVACProperties": {
          "TemperatureScales": [
            "Fahrenheit",
            "Celsius"
          ],
          "SetPointRange": {
            "Fahrenheit": {
              "Min": 32,
              "Max": 212,
              "Step": 1
            },
            "Celsius": {
              "Min": 0,
              "Max": 100,
              "Step": 0.1
            }
          },
          "FanModes": [
            "Auto",
            "High",
            "Low"
          ],
          "OperatingModes": [
            "Auto",
            "Off",
            "Heat",
            "Cool"
          ],
          "EcoModeProperties": {
            "ActiveState": "Active"
          }
        }
      },
      {
        "href": "/zone/3",
        "Name": "Hallway HVAC Zone Dual",
        "ControlType": "DualSetPointHVAC",
        "DualSetPointHVACProperties": {
          "TemperatureScales": [
            "Fahrenheit",
            "Celsius"
          ],
          "HeatingSetPointRange": {
            "F": {
              "Min": 32,
              "Max": 212,
              "Step": 1
            },
            "C": {
              "Min": 0,
              "Max": 100,
              "Step": 0.1
            }
          },
          "CoolingSetPointRange": {
            "F": {
              "Min": 32,
              "Max": 212,
              "Step": 1
            },
            "C": {
              "Min": 0,
              "Max": 100,
              "Step": 0.1
            }
          },
          "HeatingCoolingDelta": {
            "F": {
              "Min": 3
            },
            "C": {
              "Min": 1.7
          }
      },
              "Header": {
                      "Url": "/zone/5"
              }
      }
```

Master Device sends Read Response

```
{
        "CommuniqueType": "ReadResponse",
        "Header": {
                "MessageBodyType": "OneZoneDefinition",
                "StatusCode": "200 OK",
                "Url": "/zone/6"
        },
        "Body": {
                "Zone": {
                        "Name": "Island Strip Light",
                        "href": "/zone/6",
                        "ControlType": "WhiteTune",
                        "AvailableControlTypes": [
                                "WhiteTune",
                                "WarmDim"
                        ],
                        "Device": {
                                "href": "/device/4"
                        },
                        "DefaultDimCurve": "/curve/4",
                        "ColorTuningProperties": {
                                "WhiteTuningLevelRange": {
                                        "Min": 2500,
                                        "Max": 5000
                                }
                        }
                }
        }
}
```

Client sends One Zone Status Read Request:

```
{
        "CommuniqueType": "ReadRequest",
        "Header": {
                "Url": "/zone/2/status"
        }
}
```

Master Device sends Read Response:

```
{
        "CommuniqueType": "ReadResponse",
        "Header": {
                "MessageBodyType": "OneZoneStatus",
                "StatusCode": "200 OK",
                "Url": "/zone/2/status"
        },
        "Body": {
                "ZoneStatus": {
                        "Level": 0,
                        "Zone": {
                                "href": "/zone/2"
                        }
                }
        }
}
```

Zone Subscribes

Client subscribes to configuration changes on all zones.

Example Request

```
{
    "CommuniqueType": "SubscribeRequest",
    "Header": {
        "Url": "/zone",
        "Directives": {
            "SuppressMessageBody": true
        }
    }
}
```

Receive Subscribe Response

Example Response

```
{
    "CommuniqueType": "SubscribeResponse",
    "Header": {
        "StatusCode": "204 NoContent",
        "Url": "/zone",
        "Directives": {
            "SuppressMessageBody": true
        }
    }
}
```

Subsequent Asynchronous Responses will be Read Responses on Zones

Example Asynchronous Response

```
{
    "CommuniqueType": "ReadResponse",
    "Header": {
        "Url": "/zone",
        "StatusCode": "200 OK",
        "MessageBodyType": "MultipleZoneDefinition"
    },
    "Body": {
        "Zones": [
            {
                "href":"/zone/#",
                "XID":"xyz",
                "Name":"Zone 001",
                "ControlType":"Dimmed",
                "Category":{
                    "Type":"",
                    "IsLight":true
                },
                "AssociatedArea":{
                    "href":"/area/#"
                },
                "SortOrder":0
            }
        ]
    }
}
```

Commanding a Zone to a Level

GoToDimmedLevel

Integrator Type	Supported
Cert-Based	✓
Username/Password	✓

This request will only work for a Zone with an AvailableControlType of Dimmed. Attempting to send a non-Dimmed zone to a Dimmed Level will result in an exception.

If a zone doesn't support the parameters provided, the request will be rejected. At least one propery excluding FadeTime and DelayTime must be provided.

The response to this command will only contain fields that are supplied by the client. Please use a read if you'd like to get the current status of the zone, and subscribe to the zone status for async notifications.

Property	Required	Type	Details
Level		int	A percentage representing the current brightness value of the light. If omitted this will be omitted from command response. Level will be rounded to the nearest percent (no decimals).

DelayTime		Timespan	How long the system will wait before issuing the command. When omitted, a delay of 0 seconds is assumed. • Maximum DelayTime for Caséta/RA2Select: 2 hours. • Maximum DelayTime for HomeWorks/RadioRA 3: 4 hours • Minimum DelayTime for all systems: 0 hours
FadeTime		Timespan	How quickly the zone should move from the current Tuning Levels to the commanded level. When omitted, a fade of 250 milliseconds is assumed. • Maximum FadeTime for Caséta/RA2Select: 2 hours. • Maximum FadeTime for HomeWorks/RadioRA 3: 4 hours • Minimum FadeTime for all systems: 0 hours
Vibrancy		float	Explicit vibrancy value in range [0,100], can only be specified when AvailableControlTypes contains Vibrancy.
AutoVibrancy		EnabledState	Can only be specified when AvailableControlTypes contains Vibrancy.
ColorTuningStatus		ColorTuningStatus	If ColorTuningStatus is provided, you must provide only one child property.

Client sends Create Request to Command a Zone to go to a Level:

```
{
        "CommuniqueType": "CreateRequest",
        "Header": {
                "Url": "/zone/2/commandprocessor"
        },
        "Body": {
                "Command": {
                        "CommandType": "GoToDimmedLevel",
                        "DimmedLevelParameters": {
                                "Level": 100,
                 "FadeTime":"HH:MM:SS",
                 "DelayTime":"HH:MM:SS"
                        }
                }
        }
}
```

Master Device sends Create Response:

```
{
        "CommuniqueType": "CreateResponse",
        "Header": {
                "MessageBodyType": "OneZoneStatus",
                "StatusCode": "201 Created",
                "Url": "/zone/2/commandprocessor"
        },
        "Body": {
                "ZoneStatus": {
                        "href":"/zone/2/status",
                        "Level": 100,
                        "Zone": {
                                "href": "/zone/2"
                        }
                }
        }
}
```

GoToSwitchedLevel

Integrator Type	Supported
Cert-Based	✓
Username/Password	✓

This request will only work for a Zone with a ControlType of Switched. Attempting to send a non-Switched zone to a Switched Level will result in an exception.

Property	Required	Type	Details
SwitchedLevel	✓	string	Values are **"On"** and **"Off"**

DelayTime		Timespan	How long the system will wait before issuing the command. When omitted, a delay of 0 seconds is assumed. • Maximum DelayTime for Caséta/RA2Select: 2 hours. • Maximum DelayTime for HomeWorks/RadioRA 3: 4 hours • Minimum DelayTime for all systems: 0 hours

```
{
        "CommuniqueType": "CreateRequest",
        "Header": {
                "Url": "/zone/2/commandprocessor"
        },
        "Body": {
                "Command": {
                        "CommandType": "GoToSwitchedLevel",
                        "SwitchedLevelParameters": {
                                "SwitchedLevel": "On",
                                "DelayTime":"HH:MM:SS"
                        }
                }
        }
}
```

Client sends Create Request to Command a Zone to go to a Level:

Master Device sends Create Response:

```
{
        "CommuniqueType": "CreateResponse",
        "Header": {
                "MessageBodyType": "OneZoneStatus",
                "StatusCode": "201 Created",
                "Url": "/zone/2/commandprocessor"
        },
        "Body": {
                "ZoneStatus": {
                        "href":"/zone/2/status",
                        "SwitchedLevel": "On",
                        "Zone": {
                                "href": "/zone/2"
                        }
                }
        }
```

}

GoToSpectrumTuningLevel

Integrator Type	Supported
Cert-Based	✓
Username/Password	✓

This request will only work for a Zone with a ControlType of SpectrumTuning. Attempting to send a non-Spectrum Tuning zone to a Spectrum Tuning Level will result in an exception.

Property	Type	Details
Level	int	A percentage representing the current brightness value of the light. If omitted this will be omitted from command response. Level will be rounded to the nearest percent (no decimals).
Vibrancy	int	Represents the percentage of white light being created by a white LED vs the RGB LED's. Changing this field has the most effect when the zone is currently providing white light. If omitted this will be omitted from command response Vibrancy will be rounded to the nearest percent (no decimals).
ColorTuningStatus	ColorTuningStatus	Describes the color of light being provided by this zone. If omitted this will be omitted from command response
FadeTime	Timespan	How quickly the zone should move from the current Tuning Levels to the commanded level. When omitted, a fade of 250 milliseconds is assumed. • Maximum FadeTime for Caséta/RA2Select: 2 hours. • Maximum FadeTime for HomeWorks/RadioRA 3: 4 hours • Minimum FadeTime for all systems: 0 hours

DelayTime	Timespan	How long the system will wait before issuing the command. When omitted, a delay of 0 seconds is assumed. • Maximum DelayTime for Caséta/RA2Select: 2 hours. • Maximum DelayTime for HomeWorks/RadioRA 3: 4 hours • Minimum DelayTime for all systems: 0 hours

Client sends Create Request to Command a Zone to go to a Level:

```
{
        "CommuniqueType": "CreateRequest",
        "Header": {
                "Url": "/zone/2/commandprocessor"
        },
        "Body": {
                "Command": {
                "CommandType":"GoToSpectrumTuningLevel",
                "SpectrumTuningLevelParameters":{
                "Level": 5,
                "Vibrancy":75,
                "ColorTuningStatus":{
                        "HSVTuningLevel": {
                                                "Hue":55,
                        "Saturation":10
                        },
                },
                "DelayTime":"2",
                "FadeTime":"5"
                }
                }
        }
}
```

Master Device sends Create Response:

```
{
        "CommuniqueType": "CreateResponse",
        "Header": {
                "MessageBodyType": "OneZoneStatus",
                "StatusCode": "201 Created",
                "Url": "/zone/2/commandprocessor"
        },
        "Body": {
                "ZoneStatus": {
                        "href":"/zone/2/status",
                "Level": 5,
                "Vibrancy":75,
                "ColorTuningStatus":{
                "HSVTuningLevel": {
                                        "Hue":55,
                        "Saturation":10
                },
                }
                        "Zone": {
                                "href": "/zone/2"
                        }
                }
        }
}
```

GoToWhiteTuningLevel

Integrator Type	Supported
Cert-Based	✓
Username/Password	✓

This request will only work for a Zone with a ControlType of WhiteTune. Attempting to send a non-White Tuning zone to a White Tuning Level will result in an exception.

Property	Type	Details
Level	int	A percentage representing the current brightness value of the light. If omitted this will be omitted from command response. Level will be rounded to the nearest percent (no decimals).
WhiteTuningLevel	WhiteTuningLevel	Describes the desired color using the Kelvin scale, in increments of 10K. If omitted this will be omitted from command response.

FadeTime	Timespan	How quickly the zone should move from the current Tuning Levels to the commanded level. When omitted, a fade of 250 milliseconds is assumed. • Maximum FadeTime for Caséta/RA2Select: 2 hours. • Maximum FadeTime for HomeWorks/RadioRA 3: 4 hours • Minimum FadeTime for all systems: 0 hours
DelayTime	Timespan	How long the system will wait before issuing the command. When omitted, a delay of 0 seconds is assumed. • Maximum DelayTime for Caséta/RA2Select: 2 hours. • Maximum DelayTime for HomeWorks/RadioRA 3: 4 hours • Minimum DelayTime for all systems: 0 hours

Client sends Create Request to Command a Zone to go to a WhiteTuningLevel:

```
{
        "CommuniqueType": "CreateRequest",
        "Header": {
                "Url": "/zone/2/commandprocessor"
        },
        "Body": {
                "Command": {
                "CommandType":"GoToWhiteTuningLevel",
                "WhiteTuningLevelParameters":{
                "Level": 5,
                "WhiteTuningLevel":{
                        "Kelvin: 10000
                },
                "DelayTime":"2",
                "FadeTime":"5"
                }
                }
        }
}
```

Master Device sends Create Response:

```
{
        "CommuniqueType": "CreateResponse",
        "Header": {
                "MessageBodyType": "OneZoneStatus",
                "StatusCode": "201 Created",
                "Url": "/zone/2/commandprocessor"
        },
        "Body": {
                "ZoneStatus": {
                        "href":"/zone/2/status",
                "Level": 5,
                "WhiteTuningLevel":{
                "Kelvin": 10000
                },
                        "Zone": {
                                "href": "/zone/2"
                        }
                }
        }
}
```

GoToWarmDim

Integrator Type	Supported
Cert-Based	✓
Username/Password	✓

This request will only work for a Zone with a ControlType of WarmDim. Attempting to send a non-WarmDim zone to WarmDim will result in an exception.

Property	Type	Details
Level	int	A percentage representing the current brightness value of the light. If omitted this will be omitted from command response. Level will be rounded to the nearest percent (no decimals).
CurveDimming	CurveDimming	The desired WarmDim curve to send the zone to. If omitted, this will be omitted from the command response.
FadeTime	Timespan	How quickly the zone should move from the current Tuning Levels to the commanded level. When omitted, a fade of 250 milliseconds is assumed. • Maximum FadeTime for Caséta/RA2Select: 2 hours. • Maximum FadeTime for HomeWorks/RadioRA 3: 4 hours • Minimum FadeTime for all systems: 0 hours

DelayTime	Timespan	How long the system will wait before issuing the command. When omitted, a delay of 0 seconds is assumed. • Maximum DelayTime for Caséta/RA2Select: 2 hours. • Maximum DelayTime for HomeWorks/RadioRA 3: 4 hours • Minimum DelayTime for all systems: 0 hours

Client sends Create Request to Command a Zone to go to a WarmDim:

```
{
   "CommuniqueType": "CreateRequest",
   "Header": {
      "Url": "/zone/1/commandprocessor"
   },
   "Body": {
      "Command": {
         "CommandType": "GoToWarmDim",
         "WarmDimParameters": {
            "Level": 50,
            "CurveDimming": {
               "Curve": {
                  "href": "/curve/4"
               }
            },
                       "FadeTime": "5",
                       "DelayTime": "1"
         }
      }
   }
}
```

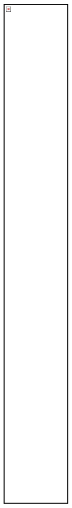
Master Device sends Create Response:

```
{
   "CommuniqueType": "CreateResponse",
   "Header": {
      "MessageBodyType": "OneZoneStatus",
      "StatusCode": "201 Created",
      "Url": "/zone/1/commandprocessor"
   },
   "Body": {
      "ZoneStatus": {
         "href": "/zone/1/status",
         "Level": 50,
                "FadeTime": "5",
                "DelayTime": "1",
         "ColorTuningStatus": {
            "CurveDimming": {
               "Curve": {
                  "href": "/curve/4"
               }
            }
         },
         "Zone": {
            "href": "/zone/1"
         },
         "StatusAccuracy": "Good",
         "Availability": "Available"
      }
   }
}
```

GoToShadeLevel

Integrator Type	Supported
Cert-Based	✓
Username/Password	✓

This request will only work for a Zone with a ControlType of Shade. Attempting to send a non-Shade zone to a Shade Level will result in an exception.

Property	Required	Type	Details	Additional Details
Level	✓	Float	The degree to which the shade is open. The semantics of the different Level values are described in this image in Additional Details.	
DelayTime		Timespan	How long the system will wait before issuing the command. When omitted, a delay of 0 seconds is assumed. Maximum delay time	

Client sends Create Request to Command a Zone to go to a Level:

```
{
        "CommuniqueType": "CreateRequest",
        "Header": {
                "Url": "/zone/2/commandprocessor"
        },
        "Body": {
                "Command": {
                        "CommandType": "GoToShadeLevel",
                        "ShadeLevelParameters": {
                                "Level": 99.75,
                    "DelayTime":"HH:MM:SS"
                        }
                }
        }
}
```

Master Device sends Create Response:

```
{
        "CommuniqueType": "CreateResponse",
        "Header": {
                "MessageBodyType": "OneZoneStatus",
                "StatusCode": "201 Created",
                "Url": "/zone/2/commandprocessor"
        },
        "Body": {
                "ZoneStatus": {
                        "href":"/zone/2/status",
                        "Level": 99.75,
                        "Zone": {
                                "href": "/zone/2"
                        }
                }
        }
}
```

GoToShadeLevelWithTilt

Integrator Type	Supported
Cert-Based	✓
Username/Password	✓

This request will only work for a Zone with a ControlType of ShadeWithTilt. Attempting to send a non-ShadeWithTilt zone to a Shade Level will result in an exception. Both level and tilt are optional but at least one must be provided in each command.

Property	Required	Type	Details	Additional Details
Level		Float	The degree to which the shade is open. The semantics of the different Level values are described in this image in Additional Details.	
Tilt		Float	The degree to which the slats are open. The semantics of the different Tilt values are described in this image in Additional Details.	
DelayTime		Timespan	How long the system will wait before issuing the command. When omitted, a delay of 0 seconds is assumed.	

Client sends Create Request to Command a Zone to go to a ShadeLevelWithTilt:

```
{
        "CommuniqueType": "CreateRequest",
        "Header": {
                "Url": "/zone/2/commandprocessor"
        },
        "Body": {
                "Command": {
                        "CommandType": "GoToShadeLevelWithTilt",
                        "ShadeWithTiltLevelParameters": {
                                "Level": 99.75,
                 "Tilt" :10.25,
                     "DelayTime":"HH:MM:SS" //Optional (1)
                        }
                }
        }
}
```

Master Device sends Create Response:

```
{
        "CommuniqueType": "CreateResponse",
        "Header": {
                "MessageBodyType": "OneZoneStatus",
                "StatusCode": "201 Created",
                "Url": "/zone/2/commandprocessor"
        },
        "Body": {
                "ZoneStatus": {
                        "href":"/zone/2/status",
                        "Level": 99.75,
                        "Tilt" : 10.25,
                        "Zone": {
                                "href": "/zone/2"
                        }
                }
        }
}
```

GoToShadeLevelWithTiltWhenClosed

Integrator Type	Supported
Cert-Based	✓
Username/Password	✓

This request will only work for a Zone with a ControlType of ShadeWithTiltWhenClosed. Attempting to send a non-ShadeWithTiltWhenClosed zone to a Shade Level will result in an exception.

Property	Required	Type	Details	Additional Details
Level	✓	Float	The degree to which the shade is open. The semantics of the different Level values are described in this image in Additional Details. **When Level is specified as non-zero, Tilt must be 0.**	
Tilt		Float	The degree to which the slats are open. The semantics of the different Tilt values are described in this image in Additional Details. **When Tilt is specified as non-zero, Level must be 0.**	
DelayTime		Timespan	How long the system will wait before issuing the command. When omitted, a delay of 0 seconds is assumed.	

Client sends Create Request to Command a Zone to go to a Level or Tilt:

```
{
        "CommuniqueType": "CreateRequest",
        "Header": {
                "Url": "/zone/2/commandprocessor"
        },
        "Body": {
                "Command": {
                        "CommandType": "GoToShadeLevelWithTiltWhenClosed",
                        "ShadeWithTiltWhenClosedLevelParameters": {
                                "Level": 0,
                 "Tilt" :10.25,
                     "DelayTime":"HH:MM:SS"
                        }
                }
        }
}
```

Master Device sends Create Response:

```
{
        "CommuniqueType": "CreateResponse",
        "Header": {
                "MessageBodyType": "OneZoneStatus",
                "StatusCode": "201 Created",
                "Url": "/zone/2/commandprocessor"
        },
        "Body": {
                "ZoneStatus": {
                        "href":"/zone/2/status",
                        "Level": 0,
                        "Tilt" : 10.25,
                        "Zone": {
                                "href": "/zone/2"
                        }
                }
        }
}
```

GoToFanSpeed

Integrator Type	Supported
Cert-Based	✓
Username/Password	✓

Property	Required	Type	Details
FanSpeed	✓	FanSpeed	Describes the speed of fan. This value determines which other properties may or may not appear in this resource.

DelayTime		Timespan	How long the system will wait before issuing the command. When omitted, a delay of 0 seconds is assumed.

```
{
   "CommuniqueType":"CreateRequest",
   "Header":{
      "Url":"/zone/3/commandprocessor"
   },
   "Body":{
      "Command":{
         "CommandType":"GoToFanSpeed",
         "FanSpeedParameters":{
            "FanSpeed":"Medium",
                       "DelayTime":"HH:MM:SS"
         }
      }
   }
}
{
        "CommuniqueType": "CreateResponse",
        "Header": {
                "MessageBodyType": "OneZoneStatus",
                "StatusCode": "201 Created",
                "Url": "/zone/3/commandprocessor"
        },
        "Body": {
                "ZoneStatus": {
                        "href":"/zone/2/status",
                        "FanSpeed":"Medium",
                        "DelayTime":"HH:MM:SS"
                }
        }
}
```

GoToCCOLevel

Integrator Type	Supported
Cert-Based	✓
Username/Password	✓

This request will work for zones that have the CCO (contact closure output) zone type

Property	Required	Type	Details
CCOLevel	✓	CCOLevelEnum	The state you would like the CCO zone to go to. Valid values: "**Open**", "**Closed**"

DelayTime		Timespan	How long the system will wait before issuing the command. When omitted, a delay of 0 seconds is assumed.

```
{
    "CommuniqueType": "CreateRequest",
    "Header": {
        "Url": "/zone/2/commandprocessor"
    },
    "Body": {
        "Command": {
            "CommandType": "GoToCCOLevel",
            "CCOLevelParameters": {
                "CCOLevel": "Open",
                "DelayTime":"HH:MM:SS"
            }
        }
    }
}
{
        "CommuniqueType": "CreateResponse",
        "Header": {
                "MessageBodyType": "OneZoneStatus",
                "StatusCode": "201 Created",
                "Url": "/zone/2/commandprocessor"
        },
        "Body": {
                "ZoneStatus": {
                        "href":"/zone/2/status",
                        "CCOLevel":"Open"
                }
        }
}
```

GoToReceptacleLevel

Integrator Type	Supported
Cert-Based	✓
Username/Password	✓

This request will work for zones that have the Receptacle zone type

Property	Required	Type	Details
ReceptacleLevel	✓	ReceptacleLevelEnum	The state you would like the Receptacle zone to go to. Valid values: "**On**" or "**Off**"

DelayTime		Timespan	How long the system will wait before issuing the command. When omitted, a delay of 0 seconds is assumed.

```
{
    "CommuniqueType": "CreateRequest",
    "Header": {
        "Url": "/zone/2/commandprocessor"
    },
    "Body": {
        "Command": {
            "CommandType": "GoToReceptacleLevel",
            "ReceptacleLevelParameters": {
                "ReceptacleLevel": "On",
                "DelayTime":"HH:MM:SS"
            }
        }
    }
}
{
        "CommuniqueType": "CreateResponse",
        "Header": {
                "MessageBodyType": "OneZoneStatus",
                "StatusCode": "201 Created",
                "Url": "/zone/2/commandprocessor"
        },
        "Body": {
                "ZoneStatus": {
                        "href":"/zone/2/status",
                        "ReceptacleLevel": "On"
                }
        }
}
```

GoToTilt

Integrator Type	Supported
Cert-Based	✓
Username/Password	✓

This request will only work for a Zone with a ControlType of ShadeWithTilt. Attempting to send a non-ShadeWithTilt zone to a Shade Level will result in an exception. Both level and tilt are optional but at least one must be provided in each command.

Property	Required	Type	Details
Tilt	✓	Float	The degree to which the slats are open.
DelayTime		Timespan	How long the system will wait before issuing the command. When omitted, a delay of 0 seconds is assumed.

Client sends Create Request to Command a Zone to go to a Level or Tilt:

```
{
        "CommuniqueType": "CreateRequest",
        "Header": {
                "Url": "/zone/2/commandprocessor"
        },
        "Body": {
                "Command": {
                        "CommandType": "GoToTilt",
                        "TiltParameters": {
                 "Tilt" :10.25,
                    "DelayTime":"HH:MM:SS"
                        }
                }
        }
}
```

Master Device sends Create Response:

```
{
        "CommuniqueType": "CreateResponse",
        "Header": {
                "MessageBodyType": "OneZoneStatus",
                "StatusCode": "201 Created",
                "Url": "/zone/2/commandprocessor"
        },
        "Body": {
                "ZoneStatus": {
                        "href":"/zone/2/status",
                        "Tilt" : 10.25,
                        "Zone": {
                                "href": "/zone/2"
                        }
                }
        }
}
```

Updating the Status of a SingleSetPointHVAC zone

Client sends an UpdateRequest

```
{
  "CommuniqueType": "UpdateRequest",
  "Header": {
    "Url": "/zone/1203/status"
  },
  "Body": {
    "ZoneStatus": {
      "SingleSetPointHvacStatus": {
        "SetPoint": {
          "F": 70
        },
        "OperatingMode": "Cool",
        "FanMode": "On"
      }
    }
  }
}
```

Master Device sends UpdateResponse

```
{
  "CommuniqueType": "UpdateResponse",
  "Header": {
    "MessageBodyType": "OneZoneStatus",
    "StatusCode": "200 OK",
    "Url": "/zone/1203/status"
  },
  "Body": {
    "ZoneStatus": {
      "href": "/zone/1203/status",
      "Zone": {
        "href": "/zone/1203"
      },
      "StatusAccuracy": "Good",
      "SingleSetPointHVACStatus": {
        "CurrentTemperature": {
          "F": 72
        },
        "SetPoint": {
          "F": 70
        },
        "OperatingMode": "Cool",
        "OperatingStatuses": [
          "CoolStageOne",
          "CoolStageTwo"
        ],
        "FanMode": "On",
        "FanStatus": "High"
      }
    }
  }
}
```

Updating the Status of a DualSetPointHVAC zone

Client sends an UpdateRequest

```
{
  "CommuniqueType": "UpdateRequest",
  "Header": {
    "Url": "/zone/1203/status"
  },
  "Body": {
    "ZoneStatus": {
```

```
□oeꟼ:s :ou}': {
```

```
      "DualSetPointHvacStatus": {
    t   "HeatingSetPoint": {: {
        □ "F": 70 "F": 76
        },
        "CoolingSetPoint": {
          "F": 76
        },
        "OperatingMode": "Cool",
   u    "FanMode": "On" : {
      }
 dniq}□ deaHer□   u  eype": "R    : {  eaHer□   u  {□{    s       pe": "T       □    □      {ies
  }
}
```

Master Device sends UpdateResponse

```
{
  "CommuniqueType": "UpdateResponse",
  "Header": {□eetunoiqueTyp{Me": "Uppe"sends"v Updat"Respo"  "r  : {  }quu  q  }
```

Zone Raise/Lower/Stop

Integrator Type	Supported
Cert-Based	✓
Username/Password	✓

Once a raise or lower is issued, the zone will continue to raise/lower until a stop command is heard. A Level Property Update for the new level will be sent after the Stop is received (reads against zone level will also continue to report the previous level until a stop is received).

Switched Zones do not support Raise, Lower or Stop commands. Use the GoToSwitchedLevel command, instead.

For SpectrumZone Raise, Lower and Stop will only affect the level property of the zone.

For non-venetian/horizontal sheer blind zones, these 3 commands will suffice. For Venetian/Horizontal Sheer Blind zones, there is also support for the following single-property commands: RaiseLift, LowerLift, StopLift, RaiseTilt, LowerTilt, and StopTilt. For more information on the behavior of these commands, see the shade zone command matrix below.

Shade Zone Command Matrix

Control Type	Raise	Lower	Stop	RaiseLift	LowerLift	StopLift	RaiseTilt	LowerTilt	StopTilt
Shade	Raises lift	Lowers lift	Stops raise/lower action	NotSupportedException	NotSupportedException	NotSupportedException	NotSupportedException	NotSupportedException	NotSupportedException
ShadeWithTilt	Raises tilt	Lowers tilt	Stops raise/lower action	Raises lift	Lowers lift	Stops lift	Raises tilt	Lowers tilt	Stops tilt

ShadeWithTiltWhenClosed	Lowers tilt to 0%, then raises lift	Lowers lift to 0%, then raises tilt	Stops raise/lower action	Raises lift (shade must be at 0% tilt)	Lowers lift	Stops lift	Raises tilt (shade must be at 0% lift)	Lowers tilt (shade must be at 0% lift)	Stops tilt

Example

Client sends Raise Command Create Request to zone :

```
{
        "CommuniqueType": "CreateRequest",
        "Header": {
                "Url": "/zone/1/commandprocessor"
        },
        "Body": {
                "Command": {
                        "CommandType": "Raise"
                }
        }
}
```

Master Device sends Command Create Response:

```
{
        "CommuniqueType": "CreateResponse",
        "Header": {
                "StatusCode": "204 NoContent",
                "Url": "/zone/1/commandprocessor"
        }
}
```

Discovering Zones in a HomeWorks/RadioRA 3 area

Integrator Type	Supported
Cert-Based	✓
Username/Password	✓

```
{
  "CommuniqueType": "ReadRequest",
  "Header": {
    "Url": "/area/757/associatedzone"
  }
}
{
  "CommuniqueType": "ReadResponse",
  "Header": {
    "MessageBodyType": "MultipleZoneDefinition",
    "StatusCode": "200 OK",
    "Url": "/area/757/associatedzone"
  },
  "Body": {
    "Zones": [
      {
        "href": "/zone/842",
        "Name": "Plug in light",
        "ControlType": "Dimmed",
        "Category": {
          "Type": "",
          "IsLight": true
        },
        "AssociatedArea": {
          "href": "/area/757"
        },
        "SortOrder": 0
      },
      {
        "href": "/zone/1368",
        "Name": "The Shade",
        "ControlType": "Shade",
        "Category": {
          "Type": "",
          "IsLight": false
        },
        "AssociatedArea": {
          "href": "/area/757"
        },
        "SortOrder": 0
      }
    ]
  }
}
```

Zone Status

Caséta	✓	
RA2 Select	✓	
HomeWorks	⚠	Read on /zone/status will not be supported. Read this to discover zone statues.

RadioRA 3	⚠	Read on /zone/status will not be supported. Read this to discover zone statues.

Integrator Type	Supported
Cert-Based	✓
Username/Password	✓

The zone status contains the current level of the zone. Different aspects of the status are available for different control types.

Property	Required	Control Type	Type	Details
href	✓	All	string	The unique identifier for this resource.
Zone	✓	All	HyperReference	A reference to the zone definition for this status.

StatusAccuracy		All	StatusAccuracyEnum	Possible values are "Good" and "Bad". • "Good" means to the best of the system's knowledge, the status reported for this zone is accurate. • "Bad" is reported when the system is unable to determine the exact values for this zone. This can happen after a power outage, and "Bad" may be reported for some zones if the loads in the zones cannot be queried for their current

				level. "Bad" does not mean the loads are offline, it simply means the system is unable to report the current status of the zoned loads. When "Bad" is reported, other status properties may be populated with default values. This property is omitted in a ReadResponse if the zone does not support this feature. StatusAccuracy should be assumed to be Good if this property is not supported.

Availability		All	AvailabilityEnum	Possible values are Available, Unavailable, and Unknown. • "Available" means that to the best of the system's knowledge, commands against this resource are likely to succeed • "Unavailable" is reported when an issue has been detected with the loads within the zone, and commands may not succeed. It is possible that even if "Unavailable" is reported, the client

				may still attempt to command the zone to a level, as the zone may have been fixed since the health of the zone was last checked. • "Unknown" is reported when the availability cannot be determined. • "Mixed" is reported when some of the commands against this resource are likely to succeed but others may not, or if a command will

				only partially succeed (e.g. if a zone contains two loads, and one of the loads is not responding, the zone status would be reported as Mixed). This property is omitted in a ReadResponse if the zone does not support this feature. Availability should be assumed to be Available if this property is not supported.
Level		Dimmed, SpectrumTune, WarmDim, WhiteTune, Shade, ShadeWithTilt, ShadeWithTiltWhenClosed	int	A percentage representing the current brightness value of the light.
SwitchedLevel		Switched		

Tilt		Shade, ShadeWithTilt, ShadeWithTiltWhenClosed	int	The tilt of the shade zone. 0-100. 50 is completely open.
FanSpeed		FanSpeed	string	The current speed of the fan. Possible values: "Off", "Low", "Medium", "MediumHigh", "High".
Vibrancy		SpectrumTune	int	Represents the percentage of white light being created by a white LED vs the RGB LED's. Changing this field has the most effect when the zone is currently providing white light.
ColorTuningStatus		SpectrumTune, WhiteTune, WarmDim	ColorTuningStatus	Describes the color of light being provided by this zone.
CCOLevel		CCOLevel	CCOLevel	
SingleSetPointHVACStatus		SingleSetPointHVAC	SingleSetPointHVACStatus	The current status of a SingleSetPointHVAC zone resource.
DualSetPointHVACStatus		DualSetPointHVAC	DualSetPointHVACStatus	The current status of a DualSetPointHVAC zone resource.

SingleSetPointHVACStatus

Property	Required	Type	Details
SetPoint	✓	SetPoint	
CurrentTemperature	✓	SetPoint	
OperatingMode	✓	OperatingMode	
OperatingStatus	✓	List<OperatingStatus>	Contains the current operating status of the HVAC zone. This field is a list to account for multiple heat/cold sources. The list will be empty if the server is not able to determine the current status.
FanMode	✓	FanMode	
FanStatus	✓	FanStatus	This is a read-only property.

DualSetPointHVACStatus

Property	Required	Type	Details
HeatingSetPoint	✓	SetPoint	
CoolingSetPoint	✓	SetPoint	
CurrentTemperature	✓	SetPoint	
OperatingMode	✓	OperatingMode	
OperatingStatus	✓	List<OperatingStatus>	Contains the current operating status of the HVAC zone. This field is a list to account for multiple heat/cold sources. The list will be empty if the server is not able to determine the current status.
FanMode	✓	FanMode	
FanStatus	✓	FanStatus	This is a read-only property.

SetPoint

Property	Required	Type	Details
F		float	The temperature in Fahrenheit
C		float	The temperature in Celsius

OperatingStatus

Value
HeatLast
HeatStageOne
HeatStageTwo
HeatStageThree
CoolLast
CoolStageOne
CoolStageTwo
Off
EmergencyHeat
Dry

FanStatus

Value
Unknown
High
Medium
Low
Off

Zone Status for Shades

```
{
        "CommuniqueType": "ReadRequest",
        "Header": {
                "Url": "/zone/1074/status"
        }
}
```

```
{
   "CommuniqueType":"ReadResponse",
   "Header":{
      "MessageBodyType":"OneZoneStatus",
      "StatusCode":"200 OK",
      "Url":"/zone/1074/status"
   },
   "Body":{
      "ZoneStatus":{
         "href":"/zone/1074/status",
         "Level":"100",
         "Tilt":"50",
         "Zone":{
            "href":"/zone/1074"
         }
      }
   }
}
```

Zone Status for Fan Speed

```
{
        "CommuniqueType": "ReadRequest",
        "Header": {
                "Url": "/zone/7/status"
        }
}
```

```
{
        "CommuniqueType": "ReadResponse",
        "Header": {
                "MessageBodyType": "OneZoneStatus",
                "StatusCode": "200 OK",
                "Url": "/zone/7/status"
        },
        "Body": {
                "ZoneStatus": {
                        "href":"/zone/7/status",
                        "FanSpeed":"Medium",
                        "Zone":{
                "href":"/zone/7"
                        }
                }
        }
}
```

Zone Status for SingleSetPointHVAC zone

```
{
  "CommuniqueType": "ReadResponse",
  "Header": {
    "MessageBodyType": "OneZoneStatus",
    "StatusCode": "200 OK",
    "Url": "/zone/5/status",
    "ClientTag": "abcxyz"
  },
  "Body": {
    "href": "/zone/5/status",
    "Zone": {
      "href": "/zone/7"
    },
    "SingleSetPointHVACStatus": {
      "SetPoint": {
        "F": 70,
        "C": 20
      },
          "CurrentTemperature": {
               "F": 70,
        "C": 20
          },
      "OperatingMode": "Auto",
      "OperatingStatuses": [
        "HeatStageOne",
        "HeatStageTwo",
        "EmergencyHeat"
      ],
      "FanMode": "On",
      "FanStatus": "Medium"
    }
  }
}
```

Zone Status for DualSetPointHVAC zone

```
{
  "CommuniqueType": "ReadResponse",
  "Header": {
    "MessageBodyType": "OneZoneStatus",
    "StatusCode": "200 OK",
    "Url": "/zone/5/status",
    "ClientTag": "abcxyz"
  },
  "Body": {
    "href": "/zone/5/status",
    "Zone": {
      "href": "/zone/7"
    },
    "DualSetPointHVACStatus": {
      "HeatingSetPoint": {
        "F": 70,
        "C": 20
      },
      "CoolingSetPoint": {
        "F": 70,
        "C": 20
      },
        "CurrentTemperature": {
               "F": 70,
        "C": 20
         },
      "OperatingMode": "Auto",
      "OperatingStatuses": [
        "HeatStageOne",
        "HeatStageTwo",
        "EmergencyHeat"
      ],
      "FanMode": "On",
      "FanStatus": "Medium"
    }
  }
}
```

Subscribing to Zone Level Updates

Integrator Type	Supported
Cert-Based	✓
Username/Password	✓

All clients currently connected will receive level property updates whenever a zone level changes in the system. These notifications appear the same as a read on a zone's status. SuppressMessageBody should always be set to true for HomeWorks and RadioRA 3 systems. Any requests without this field set will be rejected.

```
{
  "CommuniqueType": "SubscribeRequest",
  "Header": {
    "Url": "/zone/status",
       "Directives": {
      "SuppressMessageBody": true
    }
  }
}
{
  "CommuniqueType": "SubscribeResponse",
  "Header": {
    "StatusCode": "204 NoContent",
    "Url": "/zone/status",
       "Directives": {
      "SuppressMessageBody": true
    }
  }
}
```

Master Device sends (unrequested) Read Response when /zone/2's level changes:

```
{
        "CommuniqueType": "ReadResponse",
        "Header": {
                "MessageBodyType": "OneZoneStatus",
                "StatusCode": "200 OK",
                "Url": "/zone/2/status/level"
        },
        "Body": {
                "ZoneStatus": {
                        "Level": 100,
                        "Zone": {
                                "href": "/zone/2"
                        }
                }
        }
}
```

Discovering zone status of devices in an area.

Integrator Type	Supported
Cert-Based	✓
Username/Password	✓

```
{
  "CommuniqueType": "ReadRequest",
  "Header": {
    "Url": "/area/757/associatedzone/status"
  }
}
{
  "CommuniqueType": "ReadResponse",
  "Header": {
    "MessageBodyType": "MultipleZoneStatus",
    "StatusCode": "200 OK",
    "Url": "/area/757/associatedzone/status"
  },
  "Body": {
    "ZoneStatuses": [
      {
        "href": "/zone/842/status",
        "Level": 100,
        "Zone": {
          "href": "/zone/842"
        },
        "StatusAccuracy": "Good"
      },
      {
        "href": "/zone/1368/status",
        "Level": 0,
        "Zone": {
          "href": "/zone/1368"
        },
        "StatusAccuracy": "Bad"
      }
    ]
  }
}
```

Occupancy Sensors

System Availability

Caséta	✓
RA2 Select	✓
HomeWorks	✗
RadioRA 3	✗

Occupancy Sensor Resource

Occupancy sensors represent a single source of occupied/unoccupied actions. Multiple occupancy sensors may be wired to one individual device.

To subscribe to occupancy events, you should subscribe to the occupancy group that the sensor is in.

Property	Required	Type	Details
href	✓	string	The unique identifier for this resource.
Parent	✓	HyperReference	A reference to the device this occupancy sensor to which this sensor is attached.

OccupancyGroups		List<HyperReference>	The group to which this sensor belongs. The occupancy group controls the programming that will be activated when sensors in the group go occupied or unoccupied.
EnabledState		EnabledState	
OccupancySensorSettings		HyperReference	
OccupancyReportingStrategy		OccupancyReportingStrategy	

OccupancyReportingStrategy

Property	Required	Type	Details
Strategy	✓	OccupancyReportingStrategyEnum	

EnabledStateEnum

Values
Enabled
Disabled
Mixed

OccupancyReportingStrategyEnum

Values
DistributedAggregation
Independent

Occupancy Sensor Reads

Client sends a single Occupancy Sensor Read Request:

```
{
  "CommuniqueType":"ReadRequest",
  "Header":{
    "Url":"/occupancysensor/1"
  }
}
```

Master Device sends Read Response:

```
{
        "CommuniqueType": "ReadResponse",
        "Header": {
               "MessageBodyType": "OneOccupancySensorDefinition",
               "StatusCode": "200 OK",
               "Url": "/occupancysensor/1"
        },
        "Body": {
               "OccupancySensor":{
            "href" : "/occupancysensor/1"

                      "Parent": {
                             "href": "/device/2"
                      },
            "OccupancyGroups":[
               {
                 "href":"/occupancygroup/12"
               }
            ],
               }
        }
}
```

Occupancy Group

System Availability

Caséta	✓
RA2 Select	✓
HomeWorks	✗
RadioRA 3	✗

Occupancy Group Resource

An occupancy group links occupancy sensors to system programming. Whenever any sensor in the group goes occupied, the occupancy group is considered occupied and the occupied action in the programming model is triggered. Once all sensors in the group are unoccupied, the occupancy group is considered unoccupied and the unoccupied action is triggered.

Property	Required	Type	Details

href	✓	string	The unique identifier for this resource.
AssociatedAreas		List<AreaAssociation>	The area whose zones this occupancy group is able to control.
AssociatedSensors		List<AssociatedSensor>	The sensors which participate in determining the occupancy/vacancy state of this occupancy group. If one sensor is occupied, the group is occupied. If all sensors in the group are unoccupied, then the group is unoccupied.
AssociatedZones		List<AssociatedZone>	The zones that this occupancy group is able to control.
ProgrammingModel		HyperReference	
AggregatedBy		HyperReference	
ProgrammingType	✓	ProgrammingType	The property describes the programming strategy for the Occupancy Group. The only ProgrammingType currently supported is "Freeform". ▪ Freeform: Programming is applied using the ProgrammingModel

OccupancySensorAssociation

Property	Required	Type	Details
OccupancySensor	✓	HyperReference	

AssociatedZone

Property	Required	Type	Details
href	✓	string	
Zone	✓	HyperReference	

Occupancy Group Reads

Client sends All Occupancy Group Read Request:

```
{
  "CommuniqueType":"ReadRequest",
  "Header":{
    "Url":"/occupancygroup"
  }
}
```

Master Device sends Read Response:

```
{
        "CommuniqueType": "ReadResponse",
        "Header": {
                "MessageBodyType": "MultipleOccupancyGroupDefinition",
                "StatusCode": "200 OK",
                "Url": "/occupancygroup"
        },
        "Body": {
                "OccupancyGroups":[
         {
            "href" : "/occupancygroup/1",
            "ProgrammingType":"Freeform",
                       "ProgrammingModel": {
                               "href": "/programmingmodel/2"
                       },
            "AssociatedAreas":[
               {
                 "href": "/aa/5",
                                "Area": {
                                       "href":"/area/5"
                                }
               }
            ],
            "AssociatedSensors":[
               {
                  "href": "/as/6",
                                 "OccupancySensor": {
                                        "href": "/occupancysensor/13"
                                 }
               }
            ],
               },
         {
            "href" : "/occupancygroup/2",
            "ProgrammingType":"Freeform",
                       "ProgrammingModel": {
                               "href": "/programmingmodel/3"
                       },
            "AssociatedAreas":[
               {
                 "href": "/aa/6",
                                "Area": {
                                       "href":"/area/12"
                                }
               }
            ],
            "AssociatedSensors":[
               {
                  "href": "/as/5",
                                 "OccupancySensor": {
                                        "href": "/occupancysensor/12"
                                 }
               }
            ],
                }
        ]
        }
}
```

Occupancy Group Commands

Command	Arguments	Description
TestOccupiedAction	None	This will execute the programming association with this occupancy group going occupied, but it will not change the state of the occupancy group.
TestUnoccupiedAction	None	This will execute the programming association with this occupancy group going unoccupied, but it will not change the state of the occupancy group.

Client sends TestOccupiedAction command

```
{
   "CommuniqueType":"CreateRequest",
   "Url":"/og/1/commandprocessor",
   "Body":{
       "Command":{"CommandType":"TestOccupiedAction"},
   }
}
```

Master Device sends Response

```
{
   "CommuniqueType":"CreateResponse",
   "Url":"/og/1/commandprocessor",
   "Status":"204 NoContent"
}
```

Occupancy Group Status Resource

An occupancy group status is used to get runtime information on the state of the occupancy group, i.e. whether the occupancy group is currently occupied, unoccupied, or unknown.

Property	Required	Type	Details
href	✓	string	The unique identifier for this resource.
OccupancyGroup	✓	HyperReference	The occupancy group with which this status is associated.
OccupiedStatus	✓	OccupiedStatus	The current state of the occupancy group.

Occupied Status Enum

Value	Description
Occupied	At least one sensor in the occupancy group is occupied.
Unoccupied	None of the sensors in the occupancy group are reporting occupied.
Unknown	None of the online sensors are occupied, and at least one sensor is offline. This means we cannot accurately determine we are Unoccupied, so the status is instead Unknown.

Occupancy Group Status Reads

Client sends All Occupancy Group Read Request for Status:

```
{
  "CommuniqueType":"ReadRequest",
  "Header":{
    "Url":"/occupancygroup/status"
  }
}
```

Master Device sends Read Response:

```
{
        "CommuniqueType": "ReadResponse",
        "Header": {
                "MessageBodyType": "MultipleOccupancyGroupStatus",
                "StatusCode": "200 OK",
                "Url": "/occupancygroup/status"
        },
        "Body": {
                "OccupancyGroupStatuses":[
         {
            "href" : "/occupancygroup/1/status"
                       "OccupancyGroup": {
                               "href": "/occupancygroup/1"
                       },
            "OccupancyStatus":"Occupied"
                },
        {
            "href" : "/occupancygroup/2/status"
                       "OccupancyGroup": {
                               "href": "/occupancygroup/2"
                       },
            "OccupancyStatus":"Unknown"
               }
        ]
        }
}
```

Occupancy Group Status Subscription

Client sends All Occupancy Group Subscription Request for Status:

```
{
    "CommuniqueType": "SubscribeRequest",
    "Header": {
        "Url": "/occupancygroup/status"
    },
}
```

Master Device sends Successful subscription Response:

```
{
    "CommuniqueType": "SubscribeResponse",
    "Header": {
        "StatusCode": "204 NoContent",
        "Url": "occupancygroup/status"
}
```

Virtual Buttons

Caséta	✓
RA2 Select	✓
HomeWorks	✗
RadioRA 3	✗

Virtual Button Resource

The Caséta Bridge supports 50 virtual buttons. These virtual buttons, when activated, can send multiple shades or lights to a specified level, with a specified fade rate and delay. These buttons can be programmed by following the steps in Programming a Programming Model.

Property	Required	Type	Details
href	✓	HyperReference	The unique identifier for this resource.
Name		string	This is the "friendly" name of the virtual button.
ButtonNumber	✓	int	A number to uniquely identify a button on a virtual button.
ProgrammingModel		HyperReference	An href to a programming model object to give information about how the programming is structured for this virtual button.
Parent		HyperReference	The object owning this resource.
IsProgrammed	✓	bool	This property is a read-only property that is true if there are any assignments associated with the preset on this virtual button's programming model.
Category		VirtualButtonCategory	Describes the category of virtual button.
SortOrder		int	The sort order of the virtual button.

VirtualButtonCategory

Property	Required	Type	Details
Type	✓	VirtualButtonCategoryType	This value determines which other properties may or may not appear in this resource.
SubType	✓	VirtualButtonCategorySubType	This value determines which other properties may or may not appear in this resource.

VirtualButtonCategoryType

Value
Kitchen
LivingRoom
HomeAway

Bedside
AnyRoomRelax

VirtualButtonCategoryType

Value
Bright
Off
Cooking
Dining
Entertain
Movie
Relax
Home
Away
Alert
GoodNight

All Virtual Buttons

The full definition of all virtual buttons in the system.

Client sends All Virtual Button Information Read Request

```
{
        "CommuniqueType": "ReadRequest",
        "Header": {
                "Url": "/virtualbutton"
        }
}
```

Master Device sends Read Response:

```
{
        "CommuniqueType": "ReadResponse",
        "Header": {
                "MessageBodyType": "MultipleVirtualButtonDefinition",
                "StatusCode": "200 OK",
                "Url": "/virtualbutton"
        },
        "Body": {
                "VirtualButtons": [{
                        "href": "/virtualbutton/1",
                        "Name": "Button 1",
                        "ButtonNumber": 0,
                        "ProgrammingModel": {
                                "href": "/programmingmodel/1"
                        },
                        "IsProgrammed": true,
                        "Parent": {
                                "href": "/project"
                        }
                },
                {
                        "href": "/virtualbutton/2",
                        "Name": "Button 2",
                        "ButtonNumber": 1,
                        "ProgrammingModel": {
                                "href": "/programmingmodel/2"
                        },
                        "IsProgrammed": false,
                        "Parent": {
                                "href": "/project"
                        }
                },
                {
                        "href": "/virtualbutton/3",
                        "Name": "Button 3",
                        "ButtonNumber": 2,
                        "ProgrammingModel": {
                                "href": "/programmingmodel/3"
                        }"IsProgrammed": false,
                        "Parent": {
                                "href": "/project"
                        }
                }]
        }
}
```

Summarized Virtual Button Information

This is only supported for Caséta. When used in other systems it will return empty.

It's common to only want to get a list of buttons which are programmed, without getting the additional data about the names, etc.

Client sends Summary of All Virtual Buttons Read Request:

```
{
        "CommuniqueType": "ReadRequest",
        "Header": {
                "Url": "/virtualbutton/summary"
        }
}
```

Master Device sends Read Response:

```
{
        "CommuniqueType": "ReadResponse",
        "Header": {
                "MessageBodyType": "MultipleVirtualButtonDefinitionSummary",
                "StatusCode": "200 OK",
                "Url": "/virtualbutton/summary"
        },
        "Body": {
                "VirtualButtonSummaries": [{
                        "href": "/virtualbutton/1",
                        "IsProgrammed": false,
                        "Parent": {
                                "href": "/project"
                        }
                },
                {
                        "href": "/virtualbutton/2",
                        "IsProgrammed": false,
                        "Parent": {
                                "href": "/project"
                        }
                },
                {
                        "href": "/virtualbutton/3",
                        "IsProgrammed": false,
                        "Parent": {
                                "href": "/project"
                        }
                }]
        }
}
```

Single Virtual Button Information

Client sends a Virtual Button Definition Read Request:

```
{
        "CommuniqueType": "ReadRequest",
        "Header": {
                "Url": "/virtualbutton/1"
        }
}
```

Master Device send Read Response:

```
{
        "CommuniqueType": "ReadResponse",
        "Header": {
                "MessageBodyType": "OneVirtualButtonDefinition",
                "StatusCode": "200 OK",
                "Url": "/virtualbutton/1"
        },
        "Body": {
                "VirtualButton": {
                        "href": "/virtualbutton/1",
                        "Name": "Button 1",
                        "ButtonNumber": 0,
                        "ProgrammingModel": {
                                "href": "/programmingmodel/1"
                        },
                        "Parent": {
                                "href": "/project"
                        },
                        "IsProgrammed": true
                }
        }
}
```

Virtual Button Press And Release

UI Requests Virtual Button Press from Server

```
{
        "CommuniqueType": "CreateRequest",
        "Header": {
                "Url": "/virtualbutton/1/commandprocessor"
        },
        "Body": {
                "Command": {
                        "CommandType": "PressAndRelease"
                }
        }
}
```

UI waits for ACK:

```
{
        "CommuniqueType": "CreateResponse",
        "Header": {
                "StatusCode": "204 NoContent",
                "Url": "/virtualbutton/1/commandprocessor"
        }
}
```

Timestamp Resource

Property	Required	Type	Details
Day	✓	int	A day in a month. Ranges from 1 - 31

Month	✓	int	A month in a year. Ranges from 1 - 12
Year	✓	int	A year
Hour	✓	int	Ranges from 0 - 23
Minute	✓	int	Ranges from 0 - 59
Second	✓	int	Ranges from 0 - 59
Utc	✓	Timespan	Specifies the UTC offset of the timestamp.

Timespan

A string of the form "hh:mm:ss.ss", which may be preceded by a negative sign (-). Leading and trailing zeros may be omitted, but at least one digit must always appear to the left of a colon. For example, both "00:00:05.00" and "5" denote a timespan of 5 seconds, and both "02:00:00" and "2:00:00" denote a timespan of 2 hours.

TimeclockEvents

System Availability

Caséta	✓
RA2 Select	✓
Homeworks	✓
RadioRA 3	✓

Integrator Type	Supported
Cert-Based	✓
Username/Password	✓

Timeclock events describe an event in the system which is executed at specified dates and times, and with a set recurrance. When a timeclock event executes, the timeclock event's programming model is activated, causing its underlying programming to run. The execution times of a timeclock event are determined by an event's ScheduleType and TimeclockEvent type. The ScheduleType describes the date(s) that an event is executed. The TimeclockEventType describes the time of day the event is executed.

Timeclock Event Resource

Property	Required	Type	Details
href	✓	string	A unique identifier for this resource
Parent		HyperReference	The parent of the timeclock event. This is typically a Timeclock.
ProgrammingModel		HyperReference	The programming model which is activated when the event exeuctes.
Name		string	The name of the timeclock event.
ScheduleType		ScheduleType	The type of schedule that this timeclock event executes on.

Monday		bool	Indicates whether this event executes on Mondays. Only present for a DayOfWeek event.
Tuesday		bool	Indicates whether this event executes on Tuesdays. Only present for a DayOfWeek event.
Wednesday		bool	Indicates whether this event executes on Wednesdays. Only present for a DayOfWeek event.
Thursday		bool	Indicates whether this event executes on Thursdays. Only present for a DayOfWeek event.

Friday		bool	Indicates whether this event executes on Fridays. Only present for a DayOfWeek event.
Saturday		bool	Indicates whether this event executes on Saturdays. Only present for a DayOfWeek event.
Sunday		bool	Indicates whether this event executes on Sundays. Only present for a DayOfWeek event.
ExceptionDates		List<Residential Systems LEAP Protocol - Control and Monitoring#Date>	The list of dates that this event does NOT execute on. These dates do not recurr. Only present for a DayOfWeek event.

AnnuallyReocurringExceptionDates		List<Residential Systems LEAP Protocol - Control and Monitoring#Date>	The list of dates that this event does NOT execute on. These dates recurr anually. Only present for a DayOfWeek event.
BeginDate		Residential Systems LEAP Protocol - Control and Monitoring#Date	The date that this event begins evaulating days of week to execute on. Only present for a DayOfWeek event.
EndDate		Residential Systems LEAP Protocol - Control and Monitoring#Date	The date that this event stops evaulating days of week to execute on. Only present for a DayOfWeek event.
Dates		List<Residential Systems LEAP Protocol - Control and Monitoring#Date>	The list of dates that this event executes on. These dates do not recurr. Only present for a ByDate event.

AnnuallyReocurringDates		List<Residential Systems LEAP Protocol - Control and Monitoring#Date>	The list of dates that this event executes on. These dates recurr anually. Only present for a ByDate event.
TimeclockEventType		Residential Systems LEAP Protocol - Control and Monitoring#TimeclockEventType	The type of timeclock event. Indicates the time of day the event executes on.
TimeOfDay		Residential Systems LEAP Protocol - Control and Monitoring#TimeOfDay	The time of day that the event executes on. Only present for a FixedTime event.
AstronomicEventType		Residential Systems LEAP Protocol - Control and Monitoring#AstronomicEventType	The astronomic event that this timeclock event executes in occordance with. Only present for an Astronomic event.

AstronomicTimeOffset		[Residential Systems LEAP Protocol - Control and Monitoring#TimeSpan]()	The time offset to the astronomic event that the timeclock event should execute at. Can be positive or negative. Only present for an Astronomic event.

Date

Property	Required	Type	Details
Day	✓	int	The day of a month. Ranges from 1 - 31.
Month	✓	int	The month of a year. Ranges from 1 - 12.
Year		int	The year. May be omitted if date is recurring.

TimeOfDay

Property	Required	Type	Details
Hour		int	The hour of an day. Ranges from 0 - 23.
Minute		int	The minute of an hour. Ranges from 0 to 59.

ScheduleType Enum

Value	Description
DayOfWeek	Denotes a timeclock event that executes of specified days of the week (ex. Monday, Wednesday, Friday).
ByDate	Denotes a timeclock event that executes on specified dates.

TimeclockEventType Enum

Value	Description
FixedTime	Denotes a timeclock event that executes at a fixed time of day.
Astronomic	Denotes a timeclock event that executes in occordance with an astronomic event.

AstronomicEventType

Value	Description
Sunrise	Denotes an astronomic timeclock event that executes in occordance with sunrise.
Sunset	Denotes an astronomic timeclock event that executes in occordance with sunset.

Timeclock Event Read

Caséta/RA2 Select

Client requests to read all timeclock events

```
{
        "CommuniqueType": "ReadRequest",
        "Header": {
                "Url": "/timeclockevent"
        }
}
```

Processor sends back read response

```
{
    "CommuniqueType": "ReadResponse",
    "Header": {
        "MessageBodyType": "MultipleTimeclockEventDefinition",
        "StatusCode": "200 OK",
        "Url": "/timeclockevent"
    },
    "Body": {
        "TimeclockEvents": [
            {
```

```
"href": "/timeclockevent/524",
"Parent": {
    "href": "/timeclock/6"
},
"ProgrammingModel": {
    "href": "/programmingmodel/529"
},
"Name": "Timeclock Event 001",
"ScheduleType": "DayOfWeek",
"Sunday": false,
"Monday": true,
"Tuesday": true,
"Wednesday": true,
"Thursday": true,
"Friday": true,
"Saturday": false,
"ExceptionDates": [
    {
        "Day": 17,
        "Month": 2,
        "Year": 2021
    },
    {
        "Day": 19,
        "Month": 2,
        "Year": 2021
    },
    {
        "Day": 26,
        "Month": 2,
        "Year": 2021
    },
    {
        "Day": 1,
        "Month": 3,
        "Year": 2021
    },
    {
        "Day": 2,
        "Month": 3,
        "Year": 2021
    },
    {
        "Day": 5,
        "Month": 3,
        "Year": 2021
    },
    {
        "Day": 11,
        "Month": 3,
        "Year": 2021
    }
],
"TimeclockEventType": "FixedTime",
"TimeOfDay": {
    "Hour": 7,
    "Minute": 30
```

```
                }
            },
            {
                "href": "/timeclockevent/532",
                "Parent": {
                    "href": "/timeclock/6"
                },
                "ProgrammingModel": {
                    "href": "/programmingmodel/537"
                },
                "Name": "Timeclock Event 002",
                "ScheduleType": "ByDate",
                "Dates": [
                                      {
                        "Day": 2,
                        "Month": 3,
                        "Year": 2021
                    },
                    {
                        "Day": 5,
                        "Month": 3,
                        "Year": 2021
                    },
                              ],
                "TimeclockEventType": "Astronomic",
                "AstronomicEventType": "Sunset",
                "AstronomicTimeOffset": "0"
            }
        ]
    }
}
```

HomeWorks/Athena/RadioRA 3

Client requests to read single timeclock event

```
{
        "CommuniqueType": "ReadRequest",
        "Header": {
                "Url": "/timeclockevent/256"
        }
}
```

Processor sends back read response

```
{
    "CommuniqueType": "ReadResponse",
    "Header": {
        "MessageBodyType": "OneTimeclockEventDefinition",
        "StatusCode": "200 OK",
        "Url": "/timeclockevent/524"
    },
    "Body": {
        "TimeclockEvent": {
            "href": "/timeclockevent/524",
            "Parent": {
                "href": "/timeclock/6"
            },
            "ProgrammingModel": {
                "href": "/programmingmodel/529"
            },
            "Name": "Timeclock Event 001",
            "ScheduleType": "DayOfWeek",
            "Sunday": false,
            "Monday": true,
            "Tuesday": true,
            "Wednesday": true,
            "Thursday": true,
            "Friday": true,
            "Saturday": false,
            "ExceptionDates": [
                {
                    "Day": 17,
                    "Month": 2,
                    "Year": 2021
                },
                {
                    "Day": 19,
                    "Month": 2,
                    "Year": 2021
                },
                {
                    "Day": 26,
                    "Month": 2,
                    "Year": 2021
                },
                {
                    "Day": 1,
                    "Month": 3,
                    "Year": 2021
                },
                {
                    "Day": 2,
                    "Month": 3,
                    "Year": 2021
                },
                {
                    "Day": 5,
                    "Month": 3,
                    "Year": 2021
                },
                {
                    "Day": 11,
                    "Month": 3,
                    "Year": 2021
                }
            ],
            "TimeclockEventType": "FixedTime",
            "TimeOfDay": {
                "Hour": 7,
                "Minute": 30
            }
```

```
        }
    }
}
```

Timeclock Event Status Resource

Property	Required	Type	Details
TimeclockEvent		Residential Systems LEAP Protocol - Control and Monitoring#HyperReference	The timeclock event this status pertains to.
EnableState		Residential Systems LEAP Protocol - Control and Monitoring#EnabledState	The enabled state of TimeclockEvent.

Timeclock Event Status Read

Caséta/RA2 Select

Client requests to read the status of all timeclock events

```
{
    "CommuniqueType": "ReadRequest",
    "Header": {
        "Url": "/timeclockevent/status"
    }
}
```

Processor sends a read response

```
{
    "CommuniqueType": "ReadResponse",
    "Header": {
        "MessageBodyType": "MultipleTimeclockEventStatus",
        "StatusCode": "200 OK",
        "Url": "/timeclockevent/status"
    },
    "Body": {
        "TimeclockEventStatuses": [
            {
                "TimeclockEvent": {
                    "href": "/timeclockevent/524"
                },
                "EnableState": "Enabled"
            },
            {
                "TimeclockEvent": {
                    "href": "/timeclockevent/532"
                },
                "EnableState": "Enabled"
            }
        ]
    }
}
```

HomeWorks/Athena/RadioRA 3

Client requests to read a single timeclock event status

```
{
    "CommuniqueType": "ReadRequest",
    "Header": {
        "Url": "/timeclockevent/524/status"
    }
}
```

Processor sends a read response

```
{
    "CommuniqueType": "ReadResponse",
    "Header": {
        "MessageBodyType": "OneTimeclockEventStatus",
        "StatusCode": "200 OK",
        "Url": "/timeclockevent/524/status"
    },
    "Body": {
        "TimeclockEventStatus": {
            "TimeclockEvent": {
                "href": "/timeclockevent/524"
            },
            "EnableState": "Enabled"
        }
    }
}
```

Timeclock Event Status Updates

Client requests to update a timeclock event status

```
{
    "CommuniqueType": "UpdateRequest",
    "Header": {
        "Url": "/timeclockevent/524/status"
    },
    "Body": {
        "TimeclockEventStatus": {
            "EnabledState": "Enabled"
        }
    }
}
```

Processor sends an update response

```
{
    "CommuniqueType": "UpdateResponse",
    "Header": {
        "MessageBodyType": "OneTimeclockEventStatus",
        "StatusCode": "200 OK",
        "Url": "/timeclockevent/524/status"
    },
    "Body": {
        "TimeclockEventStatus": {
            "TimeclockEvent": {
                "href": "/timeclockevent/524"
            },
            "EnableState": "Enabled"
        }
    }
}
```

SystemLoadShedding

System Availability

Caséta	
RA2 Select	
Homeworks	
RadioRA 3	
Athena	

Integrator Type	Supported
Cert-Based	
Username/Password	

System loadshedding represents the state of load shedding for the entire system.

System LoadShedding Status Resource

Property	Required	Type	Details
href		string	A unique identifer for the resource
State		Residential Systems LEAP Protocol - Control and Monitoring#EnabledState	The state ofssy tattt

System Load Shedding Status Read

Client requests to read system load shedding status

```
{
    "CommuniqueType": "ReadRequest",
    "Header": {
        "Url": "/system/loadshedding/status"
    }
}
```

Processors sends read response

```
{
    "CommuniqueType": "ReadResponse",
    "Header": {
        "MessageBodyType": "OneSystemLoadSheddingStatus",
        "StatusCode": "200 OK",
        "Url": "/system/loadshedding/status"
    },
    "Body": {
        "SystemLoadSheddingStatus": {
            "href": "/system/loadshedding/status",
            "State": "Disabled"
        }
    }
}
```

System Load Shedding Status Update

Client requests to update system load shedding status

```
{
    "CommuniqueType": "UpdateRequest",
    "Header": {
        "Url": "/system/loadshedding/status"
    },
    "Body": {
        "SystemLoadSheddingStatus": {
            "href": "/system/loadshedding/status",
            "State": "Disabled"
        }
    }
}
```

Processor sends update response

```
{
    "CommuniqueType": "UpdateResponse",
    "Header": {
        "MessageBodyType": "OneSystemLoadSheddingStatus",
        "StatusCode": "200 OK",
        "Url": "/system/loadshedding/status"
    },
    "Body": {
        "SystemLoadSheddingStatus": {
            "href": "/system/loadshedding/status",
            "State": "Disabled"
        }
    }
}
```

Enumerations

DeviceType Enum

DeviceTypes reported by the Caséta Smart Bridge, Caséta Smart Bridge Pro, Ra2 Select, HomeWorks, Athena and RadioRA 3 systems.

This table is offered as a reference only. It is suggested that clients trust the Master Device to report "Unsupported" for the device type if it is actually not supported by the system. For instance, if a Caséta Smart Bridge were to report a MaestroFanSpeedController in the device list, the client should show this device in the system even though this chart suggests it is not allowed. This table captures what the system reports over leap for DeviceType field. ❌ doesn't indicate that the system doesn't support it but only indicates that system doesn't expose that information.

There is no way to tell apart a normal pico from an audio or a shade pico in a HomeWorks, Athena or RadioRA 3 system.

Value	**Caséta Smart Bridge**	**Caséta Smart Bridge Pro**	**Ra2 Select**	**HomeWorks**	**Athena**	**RadioRA3**	**Description**

Unknown	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	✓	The device is supported in the system but the system was unable to calculate the LEAPDeviceType for it. This would be a placeholder until we add support to expose the correct LEAPDeviceType for the device.
Unsupported	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	✓	The device is not compatible with this system (e.g. Perhaps the user is trying to add a Pro device to a non-Pro bridge)
SmartBridge	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	✗	
SmartBridgePro	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	✗	
RA2SelectMainRepeater	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	✗	
WirelessAuxRepeater	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	✓	
Pico1Button	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	✓	
Pico2Button	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	✓	
Pico2ButtonRaiseLower	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	✓	
Pico3Button	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	✓	
Pico3ButtonRaiseLower	✓	✓	✓	✓	✓	✓	

Pico4Button	✓	✓	✓	✓	✓	✓	
Pico4ButtonScene	✓	✓	✓	✓	✓	✓	
Pico4ButtonZone			✓	✓	✓	✓	
Pico4Button2Group			✓	✓	✓	✓	
GrafikEye				✓	✓	✓	
GrafikTHybridKeypad				✓	✓	✗	
HomeownerKeypad				✓	✓	✓	
PalladiomKeypad				✓	✓	✗	
SeeTouchKeypad				✓	✓	✓	
SeeTouchHybridKeypad				✓	✓	✓	
SeeTouchTabletopKeypad				✓	✓	✓	
SeeTouchInternational				✓	✓	✗	
HWQSProcessor	✗	✗	✗	✓	✓	✗	
QuantumProcessor	✗	✗	✗	✓	✓	✗	
myRoomProcessor	✗	✗	✗	✓	✓	✗	
WallDimmer	✓	✓	✓	✗	✗	✗	
WallSwitch	✓	✓	✓	✗	✗	✗	
PlugInDimmer	✓	✓	✓	✗	✗	✗	
PlugInSwitch	✗	✗	✓	✗	✗	✗	
MaestroFanSpeedController	✗	✗	✓	✓	✓	✓	
CasetaFanSpeedController	✓	✓	✗	✗	✗	✗	
FourGroupRemote	✓	✓	✓	✓	✓	✓	
QsWirelessShade	✗	✓	✓	✓	✓	✓	
SerenaHoneycombShade	✓	✓	✗	✗	✗	✗	
SerenaRollerShade	✓	✓	✗	✗	✗	✗	

TriathlonHoneycombShade	✓	✓	✓	✓	✓	✓	
TriathlonRollerShade	✓	✓	✓	✓	✓	✓	
KetraD3	✗	✗	✗	✓	✓	✗	
KetraD4R	✗	✗	✗	✓	✓	✗	
KetraS30	✗	✗	✗	✓	✓	✗	
KetraS38	✗	✗	✗	✓	✓	✗	
KetraA20	✗	✗	✗	✓	✓	✗	
PhantomKeypad	✗	✗	✗	✓	✓	✓	
AlisseKeypad	✗	✗	✗	✓	✓	✗	

EnabledType Enum

Value
Enabled
Disabled